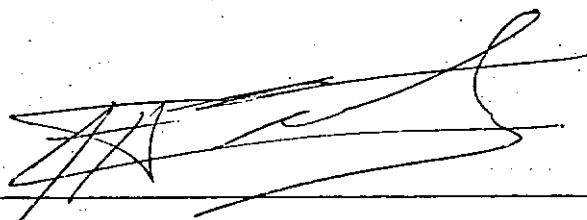


A UTILIZAÇÃO DO MINICOMPUTADOR PARA A SOLUÇÃO
DE PROBLEMAS DE ENGENHARIA CIVIL

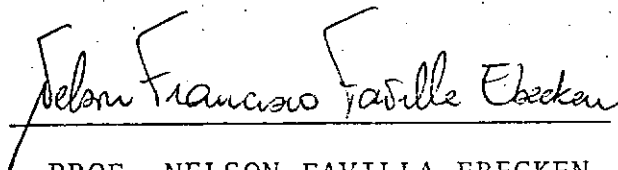
Marianela Lafuente Sanguinetti

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO
DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.)

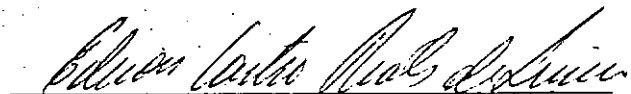
Aprovada por:



PROF. AGUSTIN JUAN FERRANTE



PROF. NELSON FAVILLA EBECKEN



PROF. EDISON C. PRATES DE LIMA

RIO DE JANEIRO, RJ - BRASIL

Julho de 1981

LAFUENTE, MARIANELA SANGUINETTI

A Utilização do Minicomputador
para a Solução de Problemas de
Engenharia Civil.

Rio de Janeiro, 1981.

VII, 236 p., 29.7cm (COPPE-UFRJ,
M.Sc. Engenharia Civil, 1981)

Tese - Universidade Federal do
Rio de Janeiro - COPPE.

1. Computação I. COPPE-UFRJ II.
Título (Série)

AGRADECIMENTOS

Agradeço aos professores e funcionários do Departamento de Engenharia da COPPE-UFRJ, que de forma direta ou indireta contribuíram na minha formação, em especial ao Professor Agustin Juan Ferrante, pela orientação, indispensável na execução deste trabalho.

Estendo ainda, meus agradecimentos, aos professores e funcionários do IMME, na Universidade Central da Venezuela, pela sua colaboração neste empreendimento.

Sou grata, finalmente, a Wilma, pela tradução e dactilografia desta tese.

SUMÁRIO

No presente trabalho é discutida a utilização de uma linguagem orientada para a resolução de problemas de engenharia, especialmente adaptada para minicomputadores, mencionando suas principais limitações e vantagens e exemplificando com alguns dos modelos mais representativos do mercado.

Em seguida, descreve-se de uma forma geral, o esquema de organização do sistema, com as características e rotinas que os integram. Apresentam-se os comandos disponíveis da linguagem orientada, as facilidades de entrada de dados e o manejo de informações, assim como os métodos de análise linear, não linear e resolução por autovalores e autovetores que se encontram implementados.

Discute-se também a estrutura interna dos sistemas, descrevendo a forma em que se realiza o manejo de informações. Enfatiza-se a descrição de uma estratégia racional, que otimiza os tempos de acesso ao disco.

Imcluem-se exemplos de aplicação, baseados na resolução de problemas governados pela equação de campo generalizada, os quais abrangem ps de torção, de hidráulica, propagação de ondas em diversos meios, etc., que são comentados e comparados com resultados conhecidos.

ABSTRACT

A problem oriented language for the solution of Engineering problems, specially prepared to be used in Minicomputers is presented.

The typical features of minicomputers are discussed first, emphasizing their limitations and advantages and describing some of the main computers types found on market.

Then, the system organization is described, including its characteristics, subroutines, main comands available, the ways of handling information and the different methods of analysis that could be used: linear and non linear analysis and eigenvector-eigenvalue solution.

The internal system structure is discussed next, describing the strategy for handling information in a rational way, in order to optimize the time of access to the disk.

Application examples are included, regarding the solution of problems that are solved by the field equation, like torsion problems, hydraulics problems, wave propagation, etc. They are comented and compared with known results.

ÍNDICE

CAP:

I. - INTRODUÇÃO	1
1.1 - OS COMPUTADORES EM ENGENHARIA CIVIL	
1.2 - O MÉTODO DOS ELEMENTOS FINITOS	4
1.3 - PLANO DE TRABALHO	6
II.- OS MINICOMPUTADORES	9
2.1 - INTRODUÇÃO	9
2.2 - CONCEITOS GERAIS	12
2.3 - CARACTERÍSTICAS GERAIS DOS MINICOMPUTADORES	17
2.4 - DESCRIÇÃO DE ALGUNS MINICOMPUTADORES NO MERCADO	27
2.5 - COMENTÁRIOS	31
III.- CARACTERÍSTICAS DO SISTEMA IMPLANTADO	33
3.1 - INTRODUÇÃO	33
3.2 - ESQUEMA GERAL	34
3.3 - COMUNICAÇÃO COM O DISCO	37
3.4 - EXPANSÕES PREVISTAS	43
IV.- APLICAÇÃO: A EQUAÇÃO DE CAMPO GENERALIZADA	46
4.1 - INTRODUÇÃO	46
4.2 - A EQUAÇÃO DE CAMPO GENERALIZADA	47
4.3 - EXEMPLOS PARTICULARES DE APLICAÇÃO	50
4.4 - FORMULAÇÃO DISCRETA PARA PROBLEMAS BASEADOS NA EQUAÇÃO DE CAMPO	59
4.5 - ELEMENTO TRIANGULAR SIMPLES	62
V. - CARACTERÍSTICAS DA LINGUAGEM	65
5.1 - INTRODUÇÃO	65
5.2 - CARACTERÍSTICAS GERAIS	66
5.3 - ALGUNS DETALHES DA IMPLEMENTAÇÃO	75
VI.- PROCEDIMENTOS DE ANÁLISE	88
6.1 - INTRODUÇÃO	88
6.2 - ANÁLISE LINEAR	88
6.3 - ANÁLISE NÃO LINEAR	90
6.4 - ANÁLISE DE PROBLEMAS DE AUTOVALORES E AUTOVETORES	93 103

VII. - UTILIZAÇÃO DO SISTEMA	116
7.1 - INTRODUÇÃO	116
7.2 - EXEMPLOS DE APLICAÇÃO PARA ANÁLISE LINEAR	116
7.3 - EXEMPLOS DE APLICAÇÃO PARA ANÁLISE NÃO LINEAR	123
7.4 - EXEMPLOS DE APLICAÇÃO PARA ANÁLISE POR AUTOVALORES E AUTOVETORES	133
7.5 - EXEMPLOS DE ENTRADAS DE DADOS	142
- CONCLUSÕES E COMENTÁRIOS	145
- BIBLIOGRAFIA	149
- APÊNDICE	153

CAPÍTULO I

INTRODUÇÃO

1.1 OS COMPUTADORES EM ENGENHARIA CIVIL

No princípio da história da computação a maioria dos homens se encontrava maravilhada pela aparição da nova máquina, o novo monstro com capacidade quase "mágica" de resolver problemas que anteriormente custavam o trabalho de anos-homem, às vezes infrutífero. O computador passou a ser um mito incompreensível para a maioria das pessoas, que viam, às vezes com terror, o advento de uma nova era e prediziam a próxima substituição do homem pela máquina, temendo e resistindo à invasão progressiva e rápida do computador em todas as áreas da vida humana.

No entanto, o computador não é mais que uma máquina, uma ferramenta do homem, sem capacidade creativa e incapaz de formular os problemas. É na realidade, um equipamento eletrônico que realiza operações matemáticas e lógicas em alta velocidade e processa grande quantidade de informação codificada. Tem-se dito que o computador é capaz de resolver qualquer problema que possa ser definido. Isto significa que a solução do problema possa ser planejada como uma série de passos que por sua vez possam ser escritos como uma sequência de instruções de computador.

Atualmente o mundo funciona quase em base de computadores: a engenharia, a aeronáutica, a economia, a medicina, a física, a estatística, quase todas as ciências devendo muito ao desenvolvimento da computação. O computador se converteu em uma ferramenta imprescindível para numerosas áreas do saber humano. Este desenvolvimento foi facilitado pela crescente disponibilidade de equipamentos de baixo custo e alta eficiência, acessíveis a pequenas firmas, indústrias, centros de pesquisas, escolas, Universidades, etc.

Este processo tem sido vertiginoso. Nos anos 50,

aparecem os primeiros computadores propriamente ditos, contruídos com base em válvulas e utilizados sobretudo para o armazenamento e manejo de bancos de dados. As aplicações na engenharia começam realmente com os computadores de segunda geração, baseados em transistores, e crescem e se reafirmam por volta de 1965, com o advento dos computadores de terceira geração, construídos em base de micro-circuitos integrados que originaram os chamados minicomputadores.

O minicomputador típico é uma máquina de baixo custo relativo, inicialmente introduzida pela "DIGITAL Equipment Corporation", de pequeno tamanho, construída para instalação simples e se é provida de um software orientado para o usuário, pode ser programada e operada por pessoas não especializadas em computadores. Por estas e outras características, vêm ganhando popularidade entre os usuários, que conseguem no mercado uma grande quantidade de equipamentos de diversas marcas.

A nova tendência é diminuir ainda mais o custo e o tamanho dos computadores. Em meados de 1979 a "Intel Corporation" introduz no mercado os chamados miniprocessadores, os "chips" integrados contendo elementos de controle e cálculo de um computador. Baseado nestes microprocessadores aparece em 1975 o primeiro microcomputador (Altair 8800), que abre o caminho aos computadores individuais.

Com a grande variedade de minicomputadores e microcomputadores de baixo custo existentes no mercado, o engenheiro vem se acercando progressivamente destas ferramentas de trabalho, utilizando-as também em sua atividade diária, para aplicações em estruturas, hidráulica, eletro-magnetismo, química, etc.

A utilidade de um computador é evidente no desenvolvimento de um modelo matemático qualquer entendendo-se este como "uma descrição da situação real em forma idealizada, adequada para o cálculo e com propósitos de predição e controle" (Benjamin e Cornell). Deve se notar que para o engenheiro, e mais especificamente para o engenheiro civil, a idealização de

um problema real é sempre dificultosa e envolve a representação de um meio contínuo. As soluções analíticas, as quais se ajustam a todos os pontos do meio estudado, só se conseguem em situações muito simplificadas. Uma solução mais realista, ao contrário, implica na utilização de métodos numéricos, os quais só oferecem resposta para um número finito de pontos selecionados sobre o meio. Aparece pois, a idéia da discretização como a única solução para numerosos problemas complexos.

No passado, o uso dos métodos numéricos esteve muito limitado pelas poucas facilidades existentes quanto à disponibilidade de computadores, ao seu alto custo e baixa capacidade. Estas limitações eram compensadas em parte pela experiência e intuição do engenheiro e pela utilização de fatores de segurança muito altos no projeto. Dentre os métodos mais utilizados estavam os de diferenças finitas, métodos de resíduos ponderados, método de Raleigh-Ritz. O método dos elementos finitos pertence totalmente à era dos computadores eletrônicos, já que nasceu por volta de 1950, graças à contribuição de Turner e colaboradores, e se estendeu rapidamente com aplicações na área da engenharia estrutural, de modo especial. O primeiro programa elaborado com base neste método foi o "STRU DL" do M.I.T., ao qual se seguiram outros de grande aceitação e enorme uso.

O método dos elementos finitos é particularmente útil para problemas envolvendo materiais não homogêneos, relações não lineares entre esforços e deformações e para a representação de condições de contorno muito complicadas. A enorme utilidade destes métodos obriga a realização de esforços que permitam sua implementação em minicomputadores, de uma forma eficiente e aproveitando ao máximo as potencialidades destas máquinas. Apesar disto, não são muitos os sistemas baseados no método dos elementos finitos que tenham sido programados para minicomputadores, tratando-se, na maioria dos casos, de versões de programas para máquinas maiores, que são adaptados aos minicomputadores.

Na Venezuela, particularmente, foram realizados

poucos esforços neste campo e os poucos programas existentes não são utilizados por falta de divulgação. As companhias privadas e as do governo preferem comprar os "pacotes" de programas elaborados no exterior e utilizá-los como verdadeiras "caixas negras" emissoras de resultados baseados nos dados fornecidos. Nas mesmas Universidades venezuelanas tem se evidenciado a necessidade de criar sistemas de programas próprios, adequados às exigências específicas das mesmas, tanto na área docente, como na de pesquisa. Não se aproveitam, portanto, as enormes facilidades e vantagens dos computadores e minicomputadores das diversas escolas e faculdades, os quais permanecem sub-utilizados e à disposição de alguns poucos usuários que realizam principalmente trabalhos individuais. Os engenheiros não são preparados para usar o computador como uma ferramenta de trabalho, nem são familiarizados com seu uso, o que limita a sua produtividade e criatividade, dificultando o desenvolvimento da engenharia nacional. Este fato, na realidade, não é isolado, mas forma parte do quadro total da engenharia venezuelana, a qual se mantém em uma posição de dependência no que diz respeito à tecnologia estrangeira e não corresponde realmente às necessidades vitais do país.

1.2 - O MÉTODO DOS ELEMENTOS FINITOS

O método dos elementos finitos requer a discretização do meio contínuo, com a definição de nós e elementos de diversas formas, configurando malhas de maior ou menor complexidade (Fig. 1.1).

Os elementos finitos podem ser bidimensionais, unidimensionais, tridimensionais, com um número variável de nós em cada caso. A aproximação do modelo sobre cada elemento se realiza com base nas chamadas funções de forma ou interpolação, as quais são geralmente polinômios. A escolha destas funções é uma etapa muito importante na resolução do problema, e envolve numerosas considerações teóricas.

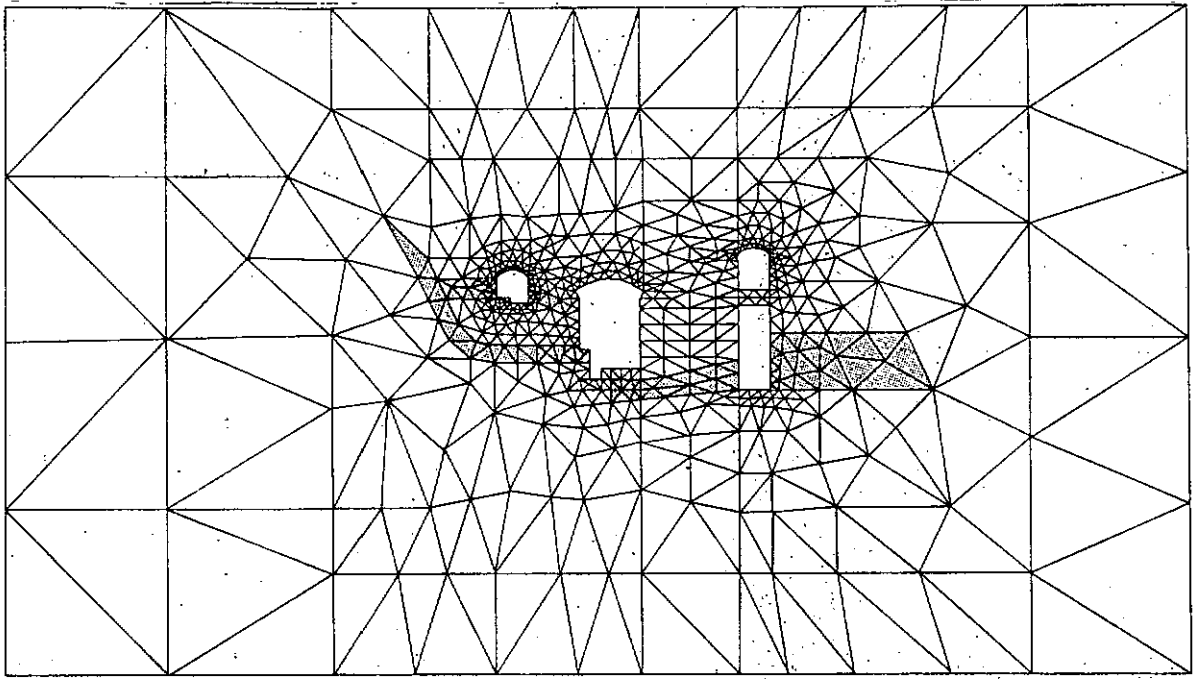


Figura 1.1
Malhas de Elementos Finitos

Na realidade, encontra-se volumosa bibliografia sobre as bases e as aplicações do método dos elementos finitos, razão pela qual neste trabalho não será incluído seu desenvolvimento teórico. Cabe, todavia, caracterizá-lo como um método aproximado baseado em princípios energéticos variacionais, cuja aplicação prática envolve as seguintes etapas.

- Discretização do modelo.
- Seleção das funções de forma.
- Construção da matriz de rigidez do elemento.
- Reunião do sistema global de equações.
- Solução do sistema.
- Cálculo de incógnitas secundárias.

É evidente que as etapas descritas são as tradicionais para a utilização de métodos matriciais na análise de estruturas de barras com a diferença de que neste caso o cálculo não implica em nenhuma aproximação para a definição das matrizes de rigidez e flexibilidade, as quais podem ser calcula-

das de forma exata baseado-se na teoria disponível. Para problemas contínuos, as matrizes de rigidez de cada elemento são aproximadas com base nas funções de forma, assim, em última instância, os resultados obtidos dependem do número de elementos empregados para discretizar o domínio de integração. A elaboração de uma malha adequada para a resolução do problema é um aspecto de vital importância e depende muito da experiência e conhecimentos prévios do usuário.

As etapas já mencionadas fazem com que este método seja facilmente adaptável ao computador e proporcionam uma estrutura adequada e geral para a análise de outros tipos de problemas, não necessariamente baseados em princípios variacionais.

Do ponto de vista computacional, este método envolve o manejo de um grande volume de dados, de acordo com a malha utilizada, e a resolução de grandes sistemas de equações. A implementação de um programa baseado no método dos elementos finitos em minicomputadores, onde a memória é limitada e o uso da memória de suporte externa leva a grandes lapsos de tempo de solução, deve ser feita levando em conta estes fatores.

Além do mais, pela grande quantidade de informação manejada, a entrada de dados se torna muito laboriosa, razão pela qual é muito útil a utilização de programas que a simplifiquem e ajudem a minimizar os erros nesta etapa. A utilização de uma linguagem orientada, além de oferecer uma entrada de dados simplificada, se adapta muito facilmente a sistemas de programas, por sua estrutura baseada em comandos que permitem a utilização independente de alguma parte dos mesmos e a seleção de características específicas para distintos tipos de problemas.

1. 3 - PLANO DE TRABALHO

Neste trabalho se descreve um sistema especialmente adaptado a minicomputadores e baseado na utilização do método dos elementos finitos. As aplicações deste sistema se rea-

lizaram na área da equação de campo generalizada, a qual abre portas não tradicionais na utilização deste método, a saber, aplicações em hidráulica, eletro-magnetismo, solos, etc. No capítulo IV descrevem-se alguns dos tipos de problemas que podem ser resolvidos através desta equação e o tipo de elementos finitos que foram empregados isto é, elementos triangulares isoparamétricos de primeira ordem. Inclui-se previamente um estudo geral das principais características dos minicomputadores, tanto em "hardware" como em "software", dando ênfase à capacidade e os custos dos principais expoentes no mercado (Capítulo II).

A organização do sistema e a forma em que se planifica é de vital importância porque dá idéia das potencialidades de expansão que apresenta e de suas facilidades de uso. Este aspecto é abordado com detalhe no capítulo III, onde se observa que o ênfase principal no sistema foi posto nas características de compatibilidade e versatilidade do mesmo. Além disso, a adaptação a minicomputadores se baseia na utilização racional do disco como memória auxiliar de suporte. Os fundamentos deste uso são descritos também neste capítulo.

A linguagem orientada é o meio de utilização do sistema por parte do usuário. Sua importância vem do fato de que se baseia em um conjunto de palavras simples, pertencentes ao vocabulário comum do engenheiro, o que facilita o uso do sistema. Sua estrutura com base em comandos isolados, facilita a integração de novas partes, ampliando sua capacidade. Seu funcionamento é descrito em detalhes no capítulo V.

O capítulo VI se ocupa em descrever os diferentes procedimentos de análise implementados no sistema, os quais permitem análises linear e não linear e de autovalores e autovetores, seguindo técnicas distintas. Dam-se, ademais, alguns detalhes de implementação, comparações e comentários gerais acerca das mesmas.

Por último, incluem-se diversos exemplos de utilização do sistema, fazendo uso dos comandos existentes na linguagem orientada. Fornecem-se comparações com resultados co-

nhécidos para dar idéia da exatidão dos obtidos (Cap. VII).

Para se encerrar o trabalho enumeram-se conclusões acerca do sistema descrito.

CAPÍTULO II

OS MINICOMPUTADORES

2.1 INTRODUÇÃO

Nas últimas décadas o mundo vem presenciando o acelerado desenvolvimento da ciência e da tecnologia nos diferentes ramos do saber humano. Novos instrumentos, novas máquinas e novas técnicas invadem o mercado constantemente; os conhecimentos adquiridos passam ao serviço do homem de formas muito diversas mudando dia a dia as características de vida e de trabalho que tradicionalmente se conheciam.

O engenheiro, como tal, não é um ente estático dentro deste processo de mudanças. Cada dia surgem novos campos de aplicação dentro de sua área, novas responsabilidades e exigências que obrigam ao máximo desenvolvimento das técnicas conhecidas e à criação de novos métodos mais eficientes e adequados que os anteriores.

Em boa parte este processo tem sido possível graças à contribuição das ciências da computação. Os computadores, são hoje em dia uma arma imprescindível na resolução de problemas das mais diversas índoles, no controle de processos de laboratórios, na elaboração de estatísticas, processamento de dados, simulação de experiências, etc.

Atualmente se conseguem no mercado, computadores de todos os tamanhos e de todos os preços, adaptados a grande quantidade de usos. A variedade se estende desde pequenas calculadoras eletrônicas programáveis, com custos até de menos de US\$200,00, até os chamados maxicomputadores que custam de US\$1.000,00 para cima, verdadeiros monstros, tanto em tamanho quanto em capacidade.

Mas nem sempre o panorama foi este. A humanidade tem dado passos gigantescos através dos séculos. As primei-

ras máquinas de calcular de que se tem conhecimento foram os "ábacos", utilizados pelos egípcios e pelos chineses muito antes da nossa era, e que ainda são usados em algumas escolas de educação elementar, para familiarizar as crianças com as operações básicas. Posteriormente apareceram diversas máquinas calculadoras do tipo mecânico como as que foram idealizadas por PASCAL, matemático francês do século XVII. Mas o verdadeiro antecessor dos gigantescos computadores do século XX aparece em 1833, graças ao cientista inglês BABBAGE, o qual projetou a chamada "máquina analítica". Esta máquina foi planejada com uma estrutura que se assemelha à das máquinas modernas, já que constava de unidades de "armazem" (memória), um módulo que Babbage chamou de "moíño", que se encarregava de executar as operações matemáticas e um "ordenador", que dirigia a ordem em que iam ser executadas estas operações. Ademais, existia um dispositivo de entrada de dados. Todavia, este processo nunca chegou a funcionar, devido à pouca precisão dos dispositivos disponíveis para esse momento.

O seguinte grande passo foi a indicação dos resultados por meio de magnitudes físicas, ao invés de quantidades matemáticas. A idéia do computador analógico surge pois, com Lord Kelvin, o qual projeta uma máquina para medir o movimento das marés. Esta idéia foi desenvolvida 50 anos mais tarde por Vannevar Bush, no Instituto Tecnológico de Massachusetts.

Em 1939, Samuel Williams, dos Laboratórios Bell, constrói o primeiro computador digital, é a primeira máquina que funciona com base no sistema binário. Já em 1944, a idéia toma corpo com o projeto da Harvard Mark I, construída pelo Professor Howard Aiken; mas é em 1946 que aparece a famosa "Eniac", gigantesco computador digital eletrônico, contendo 18000 válvulas, passando a ser a primeira máquina em seu estilo. A capacidade da ENIAC, que nessa época assombrava a humanidade, hoje nos pareceria irrisória, mas para seus criadores, J.P. Eckert e J.W. Mauchly, era quase um milagre.

Posteriormente, os computadores baseados em válvulas foram se aperfeiçoando. Na década de 1950, a IBM e

outras firmas lançam no mercado, computadores capazes de realizar 60 operações por segundo. De 1950 a 1960, as válvulas dão lugar à tecnologia dos transístores. Constroem-se memórias muito mais rápidas e de maior capacidade, em espaço físico muito menor. Aparece o conceito de circuitos integrados.

Os anos 70 vêm surgir o verdadeiro apogeu dos computadores. Os custos baixam progressivamente e a idéia de possuir uma máquina de capacidade mediana se torna possível para numerosos usuários particulares e não somente para as companhias ou entidades de grandes capitais.

Se bem que seja certo que os computadores maiores sejam mais econômicos, já que o número de operações que realizam se incrementa mais do que proporcionalmente com os custos, (Lei de GROSH) [1], também é certo que estas grandes máquinas podem não responder de uma maneira econômica a ambientes "conversacionais" para vários usuários, onde se requer rapidez na entrada e saída de informação de forma contínua, o que requer um sistema muito eficiente de tempo repartido para programas múltiplos, o que vem a aumentar muito os custos.

A maioria das máquinas grandes trabalha primordialmente em "BATCH-MODE", isto é, alimentadas de forma eficiente por uma sequência de operações a realizar, mais ou menos de forma ordenada e contínua.

Os minicomputadores surgem então, como uma solução apropriada para substituir ou complementar computadores maiores, dispondo de uma arma versátil, de baixo custo, e acessível, destinada a grupos de usuários mais restritos.

Os minicomputadores são especialmente úteis em operações onde: [1]

- a) Não se requeram circuitos muito elaborados de processador.
- b) Não haja interesse em se manter grandes e caros processadores centrais - inativos duran-

te longas operações de entrada e de saída.

O verdadeiro avanço dos minicomputadores nesta última década é devido a dois grandes fatores no desenvolvimento da tecnologia.

- a) A construção de miniprocessadores rápidos e sofisticados baseados em pequenos "chips" de silicóneo, de baixo custo.
- b) A aparição de memórias construídas com base em semi-condutores, muito mais rápidos e menos custosos que os anteriores.

Neste capítulo faz-se uma descrição das características gerais dos minicomputadores, tanto a nível de hardware como de software, referindo-se tanto à arquitetura básica de um computador, como às diferentes opções e variantes que se pode conseguir através das diversas firmas existentes no mercado.

2.2 - CONCEITOS GERAIS

Antes de se descrever as características gerais dos minicomputadores, esclarecem-se nesta secção conceitos básicos gerais os quais serão empregados posteriormente.

Os computadores, de forma geral, são máquinas organizadas pelo homem de maneira tal, que possam realizar operações repetidas em alta velocidade e de forma flexível (no sentido de poderem ser programados para resolver grande variedade de problemas) apresentando, além disso, níveis adequados de precisão.

Estas tarefas são realizadas graças a uma organização muito especial a qual está esquematizada na figura (2.1)

O conjunto integrado pela unidade de controle e pela unidade aritmética é a que comumente se conhece pelo nome de processador. [2]

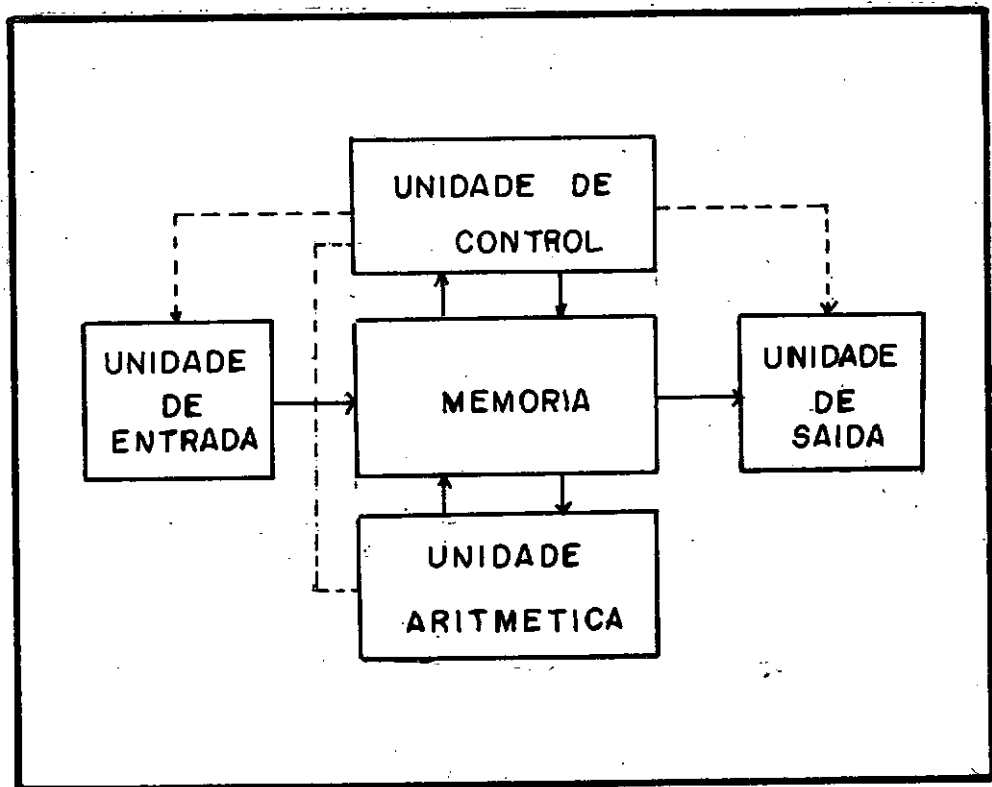


Figura 2.1
Organização de um Computador

De forma geral, a unidade de controle é a que dirige todas as operações de transferência de dados entre os registros e a memória; enquanto que as unidades de entrada e saída aceitam dados dos periféricos ou os entregam, segundo seja o caso.

A memória do computador consta de uma grande quantidade de registros direcionados, contendo informação. Adicionalmente existe uma série de circuitos que permitem selecionar uma determinada direção para ler ou escrever uma palavra.

Chama-se de "ciclo de memória" a soma do tempo de acesso a um determinado registro, previamente identificado, e sua transposição para o BUFFER, mais o tempo da escritura a partir do BUFFER em outra direção também especificada. Mais adiante se fará referência aos diferentes esquemas de direcionamento existentes.

A memória direta, ou primária, de um computador pode ser de diversos tipos:

- A chamada "core-memory", que é um tipo de memória magnética - composta por toroides de ferrita capazes de se magnetizarem em duas direções através de um impulso de corrente. Este tipo de memória apresenta a vantagem de ser não-volátil, isto é, continua armazenando seu conteúdo ainda que cesse a fonte de energia do computador.
- A memória baseada em semi-condutores, que é mais rápida e ocupa menos espaço que a anterior. O conteúdo da memória é volátil.
- Um terceiro tipo de memória substitui os toroides de ferrita por placas finas magnetizáveis. Muito rápida, não volátil, porém de alto custo.

Além da memória direta, o computador dispõe de forma de armazenamento intermediárias, que suplementam a memória principal (registros flip-flop, por exemplo) [2] e a chamada memória secundária, de baixo custo, porém mais lenta, integrada por discos magnéticos, fitas, etc., que podem armazenar milhões de palavras, transferíveis provindo da memória principal ou dirigindo-se a ela, segundo as necessidades.

Os registros da memória podem armazenar palavras de dados ou instruções. A forma de se operar com estas últimas para realizar as diferentes operações, depende do tipo de esquema de direcionamento que a máquina siga. A partir deste ponto de vista existem diversos tipos de computadores:

- As máquinas como a CDC 6600, que utiliza um esquema, como o indicado na figura (2.2), que divide cada registro em 5 partes ou setores

É evidente que se pode aumentar o tamanho das partes do registro correspondentes a direções se se elimina a divisão 5 e se substitui por um registro adicional utilizado como ponteiro e que contenha sempre a direção e a instrução em curso. Isto permite aumentar a capacidade de direcionamento de memória.

DIREÇÃO DA INSTRUÇÃO	DIREÇÃO PRIMEIRO OPERANDO	DIREÇÃO SEGUNDO OPERANDO	DIREÇÃO DO RESULTADO	DIREÇÃO PRÓXIMA INSTRUÇÃO
----------------------------	---------------------------------	--------------------------------	----------------------------	---------------------------------

Figura 2.2
Esquema de Registro

- Com o mesmo objetivo pode se eliminar também a divisão 4, utilizando a direção do segundo operando como destino do resultado. Tal é o caso de máquinas como o PDP 11, a IBM 1620, a UNIVAC 1105, etc. (máquinas de duas direções).
- Dentro das mesmas técnicas, as máquinas de uma direção eliminam outro operando substituindo-o por um registro adicional, o "acumulador". Isto explica a aparição de novas instruções que tornam mais complexa a programação, mas que aumentam a quantidade de memória direcional. Como por exemplo a PDP 8 e 15; a IBM 1130 e 7090 e a CDC 3600.
- Por último as máquinas de registros gerais possuem acumuladores múltiplos que também desempenham funções de índices em um arranjo, efetuam laços, atuam como ponteiros, etc. Exemplo: a PDP 11 e HP 1000.

Existem máquinas como a BURROUGHS 5500 e 6700 que empregam além dos registros, o conceito dos Stacks [2].

Até agora temos falado de "registros" e "palavras" sem esclarecer, todavia, o significado desses conceitos:

Como é sabido, os computadores digitais apresentam as variáveis de um problema por quantidades físicas capazes de tomar somente um número discreto de valores. A maioria imple-

menta uma representação binária em termos de variáveis básicas que podem abranger dois estados diferentes: 0 e 1. Usualmente, o estado 1 é indicado pela presença de uma voltagem em uma linha associada com a variável e o estado 0, pela sua ausência. Esta unidade de informação recebe o nome de "bit". Os bits são organizados em um registro que recebe o nome de "variável digital" ou "byte", os bytes forma um conjunto de duas ou uma unidade chamada a "palavra", que geralmente é a unidade manejada pelo usuário comum na sua programação. O byte é um registro que pode ser representado em série ou paralelo, sendo esta última representação a mais eficiente, já que permite a transmissão de todos os bits da variável ao mesmo tempo (Fig.2.3)

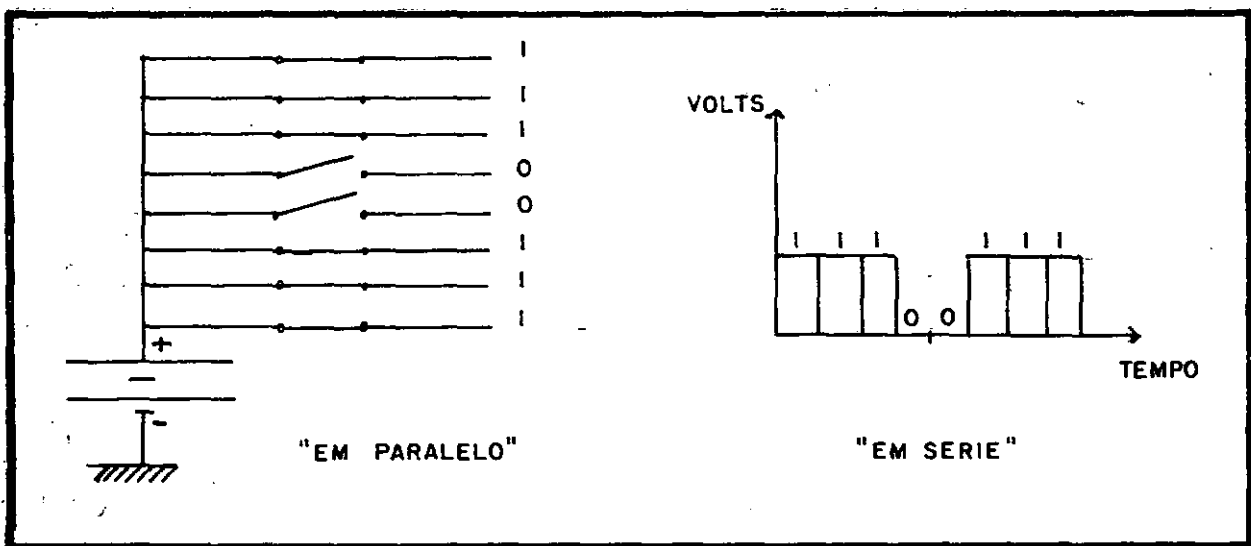


Figura 2.3
Representação de um Byte

Este tipo de representação obriga a utilização de um código binário para interpretar tanto variáveis numéricas como de qualquer outro estilo. Assim, para uma palavra de n bits o número máximo "X" a ser representado, estaria compreendido entre

$$0 \leq X \leq 2^{n-1} - 1 \quad (2.1)$$

na forma

$$X = 2^{n-1} \cdot a_0 + 2^{n-2} \cdot a_1 + \dots + a_{n-1} \quad (2.2)$$

onde a palavra com base binária correspondente a "X" é

$$(X)_{10} = (a_0, a_1, \dots, a_{n-1})_2 \quad (2.3)$$

A representação de números negativos e fracionários varia com o computador.

Para facilitar a interpretação do código binário utilizam-se os códigos octal, de base 8, ou o hexadecimal, de base 16, os quais são obtidos facilmente do anterior.

Como seria muito difícil para o programador escrever seus programas utilizando um código binário, octal ou hexadecimal, as linguagens de programação comuns realizam a comunicação com a máquina, convertendo o sistema decimal, aparente ao usuário, ao sistema interno. Estas linguagens são traduzidas para a linguagem da máquina produzindo o chamado "programa objeto". A linguagem de programação mais próxima da linguagem da máquina é a ASSEMBLER, sendo também a mais eficiente, mas é também a de mais laboriosa programação. Usualmente o Engenheiro utiliza linguagens como a FORTRAN, ALGOL, etc., dos tipos algorítmicos, e mais próximas das suas linguagens técnicas ou profissionais.

2.3 - CARACTERÍSTICAS GERAIS DOS MINICOMPUTADORES

É muito difícil definir exatamente as linhas que dividem os minicomputadores de outras máquinas maiores ou menores, e nesse sentido vêm sendo tentadas múltiplas classificações, quanto a preços, tamanho da palavra, etc. Trata-se de maxicomputadores, midicomputadores, minicomputadores, de mini-minicomputadores e de uma grande variedade de calculadoras programáveis, de bolso, etc. [11]

Neste trabalho considera-se que os minicomputadores são máquinas cujo custo oscila entre US\$ 20000 e US\$ 100000 dólares, em função do tipo e quantidade de dispositivos periféricos instalados. Em geral, a configuração mínima deste tipo de máquinas inclui umas 4000 palavras de memória direta, mas esta pode ser estendida até os 256 k bytes. Entre as exceções note-

se que a nova PDP 11/24, de baixo custo pode fazer uma memória de até 1MB. O tamanho da palavra oscila entre 8 a 18 bits, para representar dados e instruções dentro do computador.

Esta distinção dos minicomputadores quanto ao número de bits, refere-se aos registros básicos que interconectam a memória, os circuitos internos do processador e os equipamentos externos. Geralmente não se leva em conta os bits extras que se utilizam para a checagem de paridade e/ou proteção da memória. O comprimento destes registros nem sempre coincide com o comprimento dos registros aritméticos, já que estes podem combinar registros de n bits, resultando palavras compostas de $2n$, $3n$ ou $4n$ bits.

Como é evidente, o número máximo de instruções que se pode obter com uma palavra de 18 bits é 2^{18} , o qual parece uma quantidade razoável, mas deve-se levar em conta que estas instruções estão destinadas a identificar direções na memória do computador, o que em muitos casos constitui uma limitação. O projeto efetivo de um minicomputador depende muito da forma em que se aproveite o pequeno comprimento da palavra na implementação de um número reduzido de instruções capazes de realizar uma grande quantidade de operações, minimizando tanto quanto possível o incremento nos custos provocados pela utilização de instruções compostas de várias palavras.

A Tabela 2.1 mostra uma variedade de minicomputadores existentes no mercado, atendendo a uma classificação segundo o número de bits.

Pode-se afirmar que os minicomputadores possuem uma arquitetura básica comum, a saber:

- 1 - MEMÓRIA: a qual contém armazenados tanto os dados como as instruções
- 2 - REGISTROS DE PROCESSAMENTO: que incluem registros com funções de:

DIVERSOS EXEMPLOS DE MINICOMPUTADORES	
8 BITS	INTERDATA VARIAN DATA 520/i
12 BITS	SERIE PDP-8 (DEC) INC SPC-R/MOTOROLA DATA
16 BITS	HONEY-WELL 316/416/516 VARIAN DATA 620/F e L WESTINGHOUSE 2500 LOCKHEED MAC/MAC JR RAYTHON COMPUTER 704/6 HP-2100 HP-1000 PDP-11 DATA GENERAL NOVA/SUPERNOVA
18 BITS	PDP-15 VARIAN DATA 620/F GENERAL AUTOMATION SPC-16

TABELA 2.1
Exemplos de Minicomputadores

- BUFFER (contém a palavra, dado ou instrução em curso na memória)
- Indicador de direção de memória (contém a direção da localidade na memória que está sendo utilizada).
- Contador ou ponteiro da instrução a ser executada.
- Propósito geral: acumuladores, registros a

ritméticos, etc.

- Conjunto de sinais para indicar o estado das operações em curso.

- 3 - UNIDADE DE LÓGICA ARITMÉTICA: encarregada das diferentes operações aritméticas.
- 4 - CONTROLE LÓGICO: encarregado de codificar a palavra no registro de instrução, para interpretá-lo.
- 5 - CANAIS DE TRANSFERÊNCIA: ("buses") os quais se encarregam de conectar todas as unidades para a transmissão de dados entre elas.

A Figura (2.4) mostra a configuração esquematizada da PDP-11.

Muitos fabricantes de minicomputadores oferecem as seguintes opções disponíveis para melhorar e estender seu funcionamento e capacidade:

- 1 - Memória extra: que requer também a extensão do comprimento do registro de memória no processador.
- 2 - Interruptor por checagem de paridade, incluindo um bit extra por palavra e toda a lógica correspondente.
- 3 - Proteção de áreas selecionadas de memória.
- 4 - Proteção para o caso de uma falha na potência.
- 5 - HARDWARE para suprir a aritmética de ponto flutuante.
- 6 - Carregadores automáticos de programas em fita de papel magnética.

Muitas outras opções consistentes em melhoras tanto do hardware como do software, podem ser conseguidas em alguns computadores.

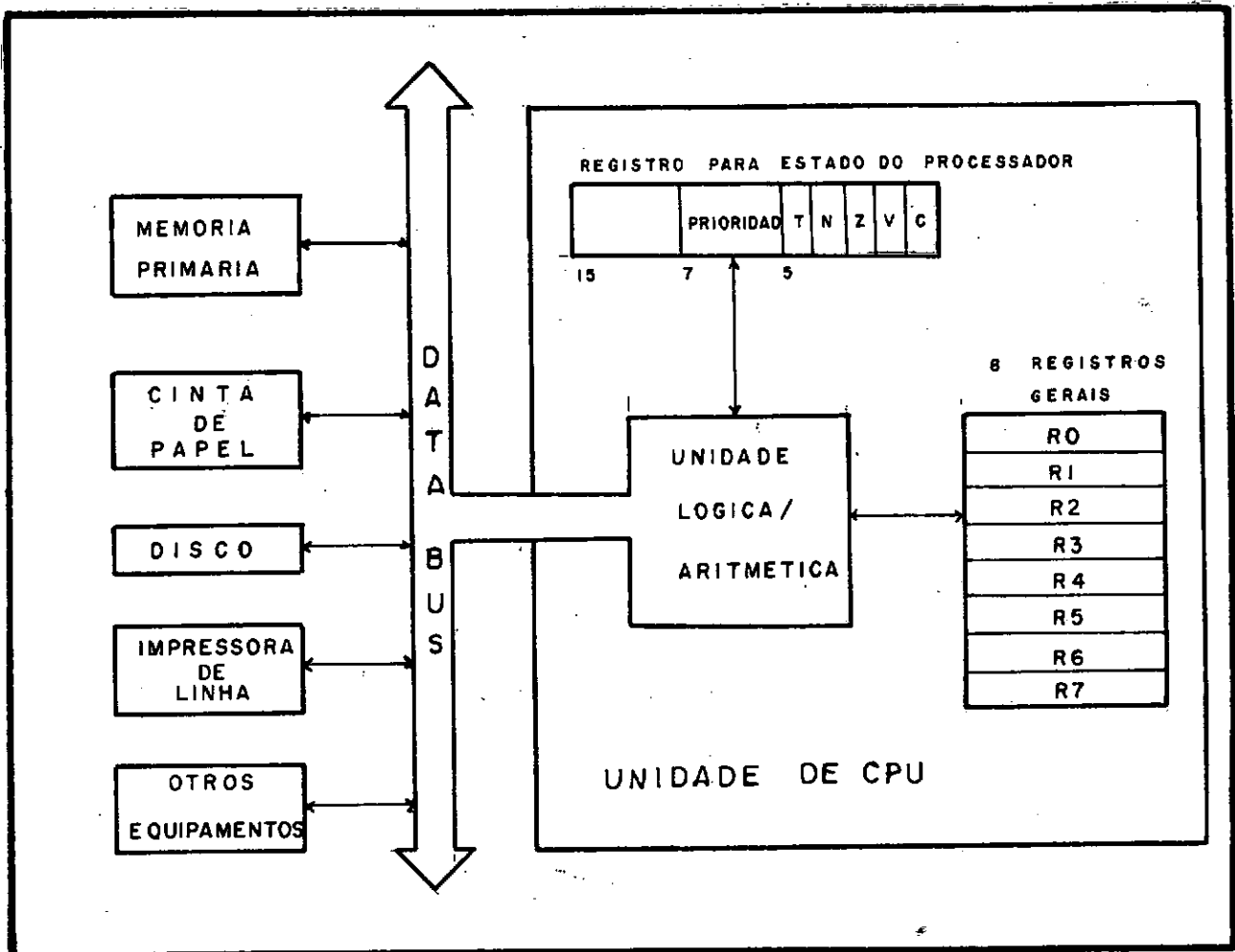


Figura 2.4
Organização e Estrutura
da PDP-11

As características gerais são esquematizadas na Ta
bela 2.2.

CARACTERÍSTICAS GERAIS DOS MINICOMPUTADORES	
TAMANHO DA PALAVRA	8 - 18 BITS
TEMPO DE UM CICLO DE MEMÓRIA	300 nseg a 4 Mseg
TAMANHO DA MEMÓRIA	0,5 - 256 Kbytes
NÚMERO DE REGISTROS	1 - 16
TRANSFERÊNCIA DIRETA PARA A MEMÓRIA	0,2 - 2 MHZ
NÚMERO DE INTERRUPTORES EXTERNOS	1-256
OPÇÕES TÍPICAS A NÍVEL DE HARDWARE	CHECAGEM DE PARIDADE PROTEÇÃO DA MEMÓRIA HARDWARE PARA PONTO FLUTUANTE, ETC
SOFTWARE RECOMENDÁVEL	FORTRAN IV e/ou ALGOL; BASIC, SISTEMA DE OPERAÇÃO DE DISCO TEMPO REAL EDITOR LOADER ASSEMBLY

TABELA 2.2

2.3.1 Características Gerais de Software

O software que sustenta o sistema de um minicomputador descansa sobretudo no programa executivo ou programa monitor, que se encarrega do trabalho de ler e armazenar programas fonte e objeto, de carregá-los e executá-los automaticamente ou baseado em comandos supridos pelo operador. Isto combina o uso de programas dos sistemas (LOADERS, ASSEMBLER, COMPILER) o manejo de sistemas de armazenamento (discos, fitas, etc.) o programa de livreria e arquivos próprios do usuário, etc.

A forma em que se concebe o sistema operativo ou programa monitor é fundamentalmente importante para a eficiência de todo o sistema. Existem esquemas idealizados para o usuário simples e esquemas que permitem a multiprogramação, quer seja através da divisão da memória ou do tempo dividido do processador.

O fabricante geralmente fornece um conjunto de sub-rotinas para as operações de entrada e saída, o que facilita ao usuário o manejo dos equipamentos.

O uso da memória de suporte de baixo custo, como discos e fitas, torna-se mais simples para o usuário pela presença de programas que transferem a informação à memória direta. O programador simplesmente deve conhecer as áreas de BUFFER, nesta memória, que conterão os blocos de dados a serem transferidos.

A programação se realiza numa linguagem como a assembler, mais próxima da linguagem da máquina, ou em alguma outra suscetível de ser traduzida pelo compilador: FORTRAN, ALGOL etc, as quais são mais parecidas com a forma de trabalho normal. Na realidade a linguagem que leva maior vantagem da configuração do hardware do minicomputador é o Assembler, o qual, sendo bem utilizado, permite economizar memória e tempo de execução, sendo que os outros, que requerem um compilador, são mais lentos.

Muitos minicomputadores fornecem linguagens que permitem um modo de trabalho conversacional. Destes, o mais popu-

EQUIPAMENTO	CUSTO APROXIMADO
PROCESSADOR + 64 K DE MEMÓRIA	US\$ 10000
HARDWARE PARA MULTIPLICAR E DIVIDIR	US\$ 1500
64 K EXTRA DE MEMÓRIA	US\$ 2000
128 K ADICIONAIS DE MEMÓRIA	US\$ 4000
UNIDADES DE CASSETE	US\$ 2000
LEITORA DE FITA DE PAPEL	US\$ 2600
LEITORA DE CARTÕES	US\$ 700
DISCO DE CABEÇA FIXA (500 kb)	US\$ 10000
IMPRESSORA DE LINHA	US\$ 8000
IMPRESSORA TERMICA	US\$ 3500
PLOTTER	US\$ <5000

TABELA 2.3
Custos Típicos
(Minicomputadores de 16 bits)

lar é o BASIC, ainda que tenham sido desenvolvidas linguagens similares como a FOCAL, da DIGITAL EQUIPMENT CORPORATION. Estes interpretadores facilitam a aritmética do ponto flutuante, rotinas próprias permitindo operações com matrizes e fornecendo funções trigonométricas, saídas gráficas, etc.

2.3.2 - Comentários Adicionais sobre o Hardware

Muitos computadores fornecem dispositivos de hardware, que suprem o computador com aritmética de ponto flutuante, operações de multiplicação e divisão, acesso à memória direta, etc.

Os minicomputadores usualmente transmitem dados através de linhas de conexão de palavras de 8 a 18 bits, que recebem o nome de "buses", os quais podem ser bidirecionais, de entrada ou saída. Para o controle da interfase se implementam então, diversos sistemas de prioridades e de interruptores.

Na maioria dos casos utiliza-se hardware adicional para minimizar a intervenção do processador, comunicando diretamente o bus de dados com a memória. Em algumas máquinas, inclusive, como a PDP-11, o bus de dados pode ter acesso a um banco de memória sem interromper a interação do processador com outros bancos. Isto é, a intervenção do processador se torna nula.

As facilidades disponíveis quanto à memória de suporte externa, incluem fitas magnéticas, discos flutuantes, discos fixos e móveis. Destes, as primeiras têm maior capacidade e mais baixa velocidade; os discos são mais custosos mas também mais rápidos, variando a sua capacidade de armazenamento de uns 512 K byte para FLOPPY DISK a uns 300 M bytes para discos de cabeça móvel.

Quanto a terminais e equipamentos periféricos, é muito utilizado o terminal de vídeo, tipo (CRT/KEYBOARDS TERMINAL) baseado em tubos de raios catódicos, complementando com impressores de linha, leitoras de cartões, plotters, etc.

2.3.3 - A Microprogramação

Merece um comentário especial esta técnica introduzida por Wilkes em 1951, da Universidade de Manchester, já que tem sido amplamente utilizada como uma possibilidade para adaptar o software aos grandes avanços do hardware.

As micro-operações se obtêm em um minicomputador a partir da descodificação de uma instrução nas diferentes partes que a compõem (combinações de bits). Uma unidade de controle microprogramada obtém estas micro-operações e as elabora em uma memória de controle, atuando assim como um "computador interno" de programas armazenados (micro-processador).

A importância da micro-programação se apoia no fato de que, com um conjunto bem escolhido de micro-instruções pode-se mudar a vontade o conjunto de instruções do computador, o que permite imitar com uma mesma estrutura, diferentes computadores para usar com diferentes aplicações de software.

A microprogramação também é útil quando se requer implementar um conjunto complexo e grande de operações (que requeririam a elaboração de subrotinas em computadores convencionais) com formas de processamento mais simples e funcionamento rápido. Tal é o caso da implementação de aritmética de ponto flutuante, rotinas para funções trigonométricas, transformadas de Fourier, integrações, operações de entrada-saída, etc.

Para simplificar a micro-programação e torná-la acessível ao usuário, o fabricante fornece uma linguagem "micro-assembler" e outras facilidades.

Os primeiros minicomputadores que apareceram no mercado foram: os modelos INTERDATA 3,4,5 e 70, e MICRODATA 810, como minicomputadores microprogramados.

Posteriormente, foi-se generalizando a tendência de se implementar conjuntos sofisticados de instruções nos computadores através da microprogramação. No entanto o custo

para conseguir um software adequado à grande capacidade facilitada pela micro-programação é excessivo, pelo que se observa que muitos minicomputadores novos, microprogramados, são colocados no mercado com sistemas de software mais antigos. Tal é o caso, por exemplo, do HP-2100 e mais recentemente do HP-1000.

2.4 - DESCRIÇÃO DE ALGUNS MINICOMPUTADORES NO MERCADO

Máquinas de 8-bits:

São amplamente usadas no controle de comunicações e como processadores periféricos para computadores maiores. Por isto, seu projeto dá ênfase à agilização das operações de entrada-saída.

Exemplos: INTERDATA 1 e VARIANDATA 520/i

Máquinas de 12-bits:

Alcançaram grande êxito nos anos 70, através de expoentes como a série PDP-8 da Digital, devido à grande quantidade de software, que foi desenvolvido para estas máquinas, destinado a grande quantidade de usos e operações.

Outro exemplo desta categoria é a INC SPC-12/MOTOROLA DATA.

Máquinas de 16 a 18-bits

Nestas máquinas o projeto do processador atende a necessidades mais complexas que para os casos anteriores.

Exemplos para 16 bits:

- Honey-well 316/416/516
- Varian Data 620/F e L
- Westinghouse 2500
- Lockheed MAC/MAC JR.
- Ruy Theon Computer 704/6
- HP-2100
- HP-1000
- PDP-11
- MICRONOVA, DATA GENERAL

(considerada pelo seu custo como micro-computador,

porém com muitas das capacidades de um mini-computador).

- DATA GENERAL, NOVA/SUPERNOVA

Exemplos para 18 bits:

Cabe comentar que estas máquinas apresentam vantagens em aplicações gráficas e de plotter, já que as coordenadas X e Y são representadas por conjuntos de 9 bits e podem ser combinadas em uma só palavra. Entre algumas destas máquinas estão:

- VARIAN DATA 620/F
- GENERAL AUTOMATION SPC-16
- PDP-15

A seguir são descritos alguns dos equipamentos mencionados que tiveram maior aceitação no mercado internacional, fazendo breve referência a suas principais características.

DIGITAL EQUIPMENT CORPORATION: PDP-11

Nesta série se observam notáveis melhoras na PDP-8, tanto no projeto do processador, como nas operações e aplicações de sistemas completos, envolvendo periféricos e processadores múltiplos.

Contém 8 registros, suscetíveis de serem utilizados como acumuladores. Todas as transferências de dados entre memória, processador e periféricos se realizam através de um único conjunto de linhas chamado "UNIBUS", que simplifica a arquitetura do conjunto, mas restringe a velocidade de operação. Em alguns sistemas como a PDP-11/45, aumenta-se a velocidade de transferência com buses adicionais de acesso rápido entre memória e processador (tempo de acesso = 300 nanosegundos). Torna-se possível o acesso a 124 k bytes de memória para aplicações de multi-programação e tempo dividido. A memória básica é de 4096 palavras de 16 bits.

Apesar de que o sistema de conexão UNIBUS seja

de baixa velocidade, esta desvantagem é compensada pela flexibilidade e versatilidade do projeto, sua fácil ampliação mediante a adição de novos módulos e seu baixo custo. Com este esquema se facilita o uso de processadores múltiplos, através de um sistema adequado de prioridades. Neste caso também pode-se possibilitar o uso de UNIBUSES múltiplos para uso de vários processadores. (ver Fig.2.5).

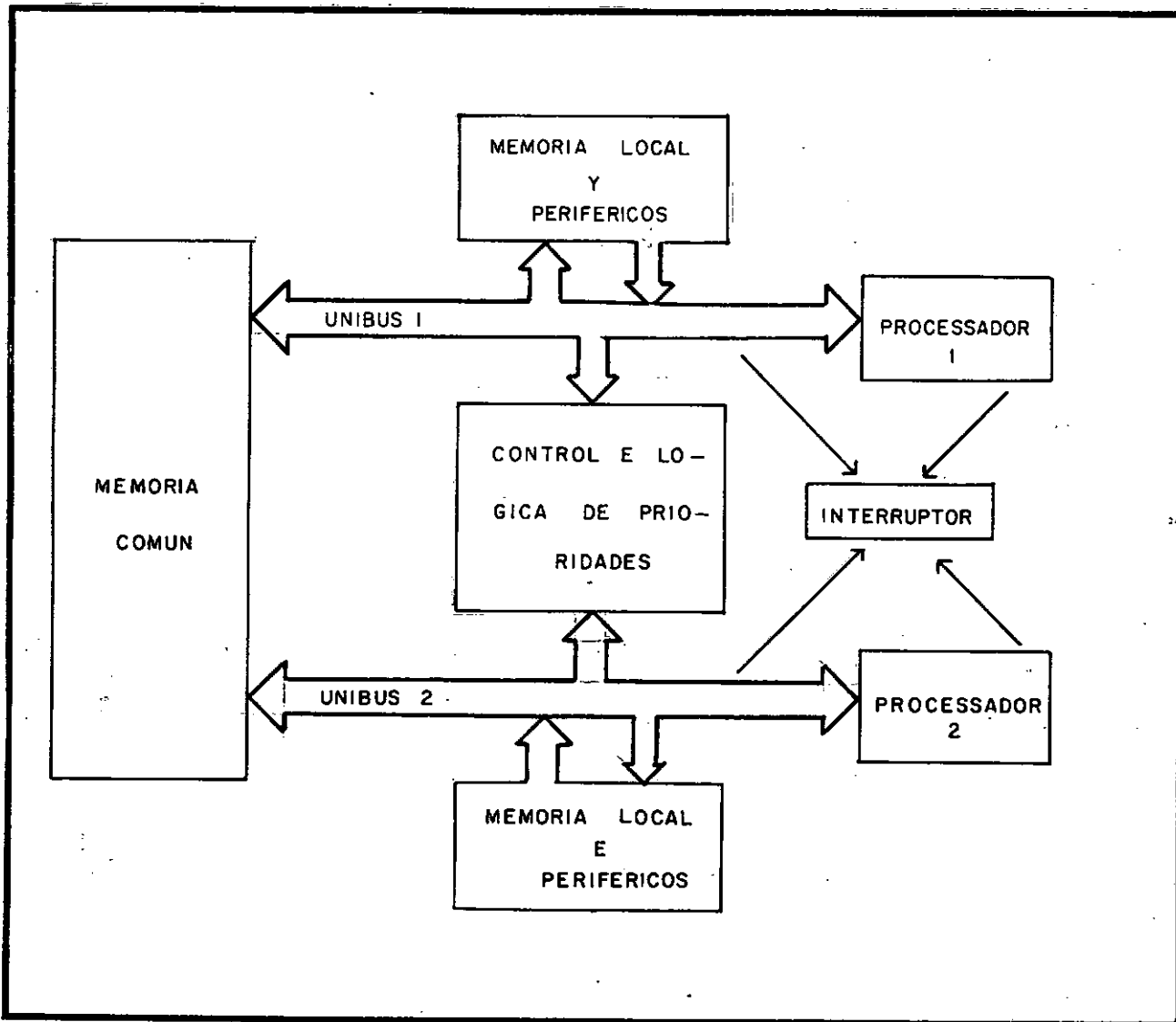


Figura 2.5
Esquema de Funcionamento de um
Sistema de Multi-Processadores

DATA GENERAL MICRONOVA/NOVA/SUPERNOVA

São uma série de computadores de 16 bits com diferentes velocidades de memória, diferentes custos e que compartilham de um mesmo conjunto de instruções e software.

O projeto compacto do processador oferece vantagens no custo, mas podem requerer mais memória que outros tipos de computador, dependendo do tipo de aplicação em que seja empregado.

Para dar uma idéia do preço destas máquinas, a micronova a um custo de aproximadamente US\$ 13500, inclui o processador, 64 k bytes de memória, sistema de diskette duplo (315 k words) sistema de operação do disco, terminal com display de vídeo, impressora, carregador automático de programas (LOADER) e algumas outras opções.

HP-2100S

Melhora as máquinas anteriores (HP-2114, 2115, 2116, 2100 A) sobretudo no que se refere ao hardware. Direciona até 16384 palavras de 16 bits em memória. (Um bit adicional por palavra para controle de paridade)

Estas máquinas possuem 8 registros múltiplos e 7 registros para micro-processadores. O ciclo de memória é de 980 nanosegundos, para uma memória de tipo magnético. Funciona com sinais para a proteção da memória e operações de I/O, com acesso direto à memória e com operações aritméticas de ponto flutuante. O sistema operativo descansa no chamado RTE, incluindo diversos sistemas de controle.

Os periféricos disponíveis incluem fita magnética, discos de cabeça fixa (262144 a 1048576 palavras) e discos móveis (1,2 milhões a 4,8 milhões de palavras).

SERIE HP-1000 E

Com mais capacidade que a série anterior, possui um

processador micro-programado que inclui: funções aritméticas, operações de entrada e saída e um painel operador de controle. Oferece ao usuário múltiplas facilidades para a micro-programação.

Dispõe de 4 registros de propósito geral e um total de 128 instruções standards para aritmética de ponto flutuante, operações de I/O, operações lógicas e manipulação a nível de bytes e de bits.

A memória direta está baseada em chips de semi-condutores, o que permite ciclos de memória de 350 n seg. Os módulos podem ser estendidos até os 128 k bytes, e com um sistema de mapeamento dinâmico (DMS) permite aos usuários direcionar até 2 bytes de memória. (ver Tabela 2.4)

O software disponível também se baseia no RTE (sistemas multi-ptogramados que permitem incluir vários programas em tempo real, enquanto se realizam outros processos concorrentes de "background"). Possui compiladores de HP FORTRAN IV, HP BASIC e um assembler muito eficiente.

2.5 - COMENTÁRIOS:

Neste capítulo foram revisadas as características típicas dos computadores com as facilidades que oferecem para o usuário. No entanto a compra de um minicomputador deve ser decidida após um estudo das necessidades que este deva cobrir, e as possibilidades de manutenção, expansão, etc., que o façam mais eficiente e econômico para o comprador.

TIPO	SÉRIE M			SÉRIE E			SÉRIE F
MODELO DO COMPUTADOR	2105 A	2108 M	2112 M	2109 E	2113 E	2111 F	2117 F
VELOCIDADE DE MEMÓRIA	650 nseg	650 nseg	650 nseg	595 nseg 350 nseg	595 nseg 350 nseg	350 nseg	350 nseg
MAXIMA MEMÓRIA EM ESTRUTURA PRINCIPAL	64 kb	64 kb	128 kb	64 kb	128 kb	64 kb	128 kb
CAPACIDADE DE EXTENSÃO DE MEMÓRIA	-	1152 kb	768 kb	1152 kb	768 kb	1,8 Mb	1,8 Mb
CANAIS DE I/O	36	41	46	41	46	46	46

TABELA 2.4

Algumas Características para

Minicomputadores HP-1000

CAPÍTULO III

CARACTERÍSTICAS DO SISTEMA IMPLANTADO

3.1 INTRODUÇÃO

Neste capítulo são descritas as características do sistema implantado. Estas características respondem a uma filosofia modular que permite a expansão e adição de novos tipos de análise de forma independente, sem alterar os já existentes.

A idéia é permitir a criação de um sistema que, contando com suficientes rotinas de serviço (resolução de sistemas de equações, montagem de matrizes, etc.) seja capaz de realizar diferentes tipos de análises (análise não linear, análise dinâmica, análise de vibrações livres, etc.) para um número considerável de problemas de aplicação (Equação de campo generalizada, problemas planos de elasticidade, problemas de estrutura de barras, etc.)

O conjunto integrado de todos estes módulos tornará possível a resolução de um número considerável de problemas de engenharia, e fornecerá um sistema versátil, baseado no método dos elementos finitos e adaptado para minicomputadores. O problema de armazenamento no disco está completamente resolvido para o usuário, o qual dispõe de suficiente memória para resolver problemas de tamanho considerável.

Outra das características de todas as partes do sistema, é que se encontram escritas em um FORTRAN altamente compatível, que permite sua implementação em qualquer máquina com extrema facilidade. Isto é devido ao fato de que todas as zonas de incompatibilidade se encontram juntas e se conhece a sua colocação (por exemplo, as rotinas que escrevem ou lêem o disco).

Este sistema constitui um projeto bastante ambicioso, o qual foi concebido como uma tarefa a longo prazo. Neste trabalho se desenvolve uma primeira etapa do mencionado sistema, a qual cumpre com os objetivos de proporcionar um corpo de pro-

gramas flexível disponível para realizar aplicações da equação de campo, na área de análise linear, análise não linear, e de autovalores e autovetores. No entanto, sua versatilidade permitirá expandi-lo com facilidade, contando-se para isso com os módulos de serviços existentes, que incluem rotinas de montagem, das matrizes de massa e rigidez, rotinas para solução de sistemas de equações por triangularização de Gauss, rotinas de Jacobi Generalizado e rotinas para resolução por iteração por sub-espacos.

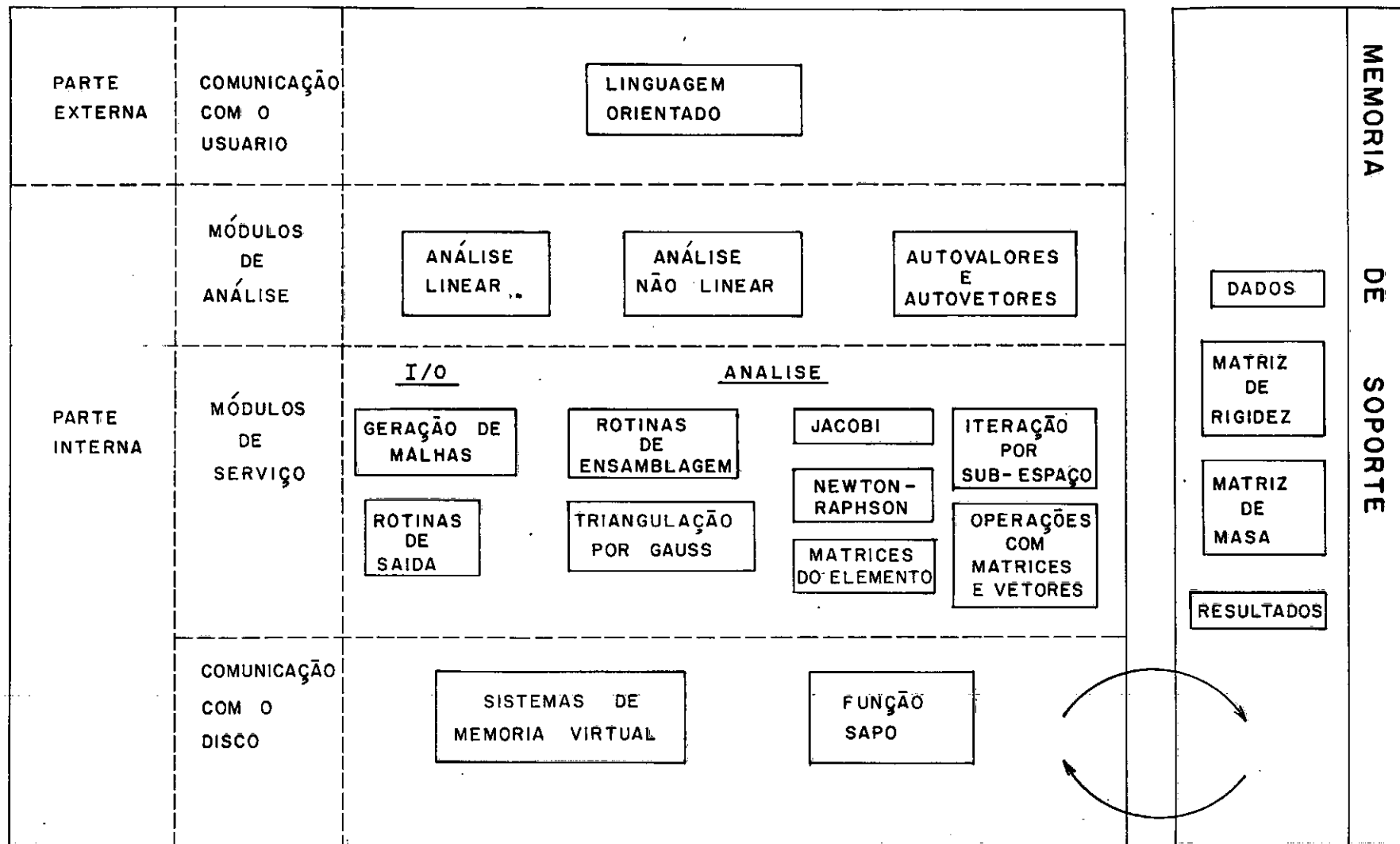
Neste capítulo se descreve o esquema geral de funcionamento do sistema, e se faz referências às expansões que se projetam tanto na parte externa (de comunicação com o usuário) como a nível de funcionamento interno do mesmo (módulos de análises e serviço).

3.2 - ESQUEMA GERAL

O estado atual do sistema é representado na Fig. 3.1. Foram divididas as diferentes partes de uma zona externa, ou de comunicação com o usuário ao nível da entrada e saída de dados e resultados e da organização dos mesmos, e uma zona interna, ou propriamente de funcionamento e análise desses dados.

A linguagem orientada constitui o sistema de comunicação com o usuário, e é encarregada de organizar os dados e armazenar a informação necessária acerca do tipo de análise a se realizar e de que forma se irá apresentar a saída de resultados. Tanto os comandos da linguagem orientada como as matrizes de rigidez para os elementos, estão orientados para a resolução de problemas de aplicação da equação de campo generalizada. Esta equação e sua formulação matricial para problemas planos se encontram desenvolvidas no capítulo IV. Assim sendo, o funcionamento geral da linguagem orientada pode ser estudada no capítulo V.

Os dados são armazenados em uma organização conveniente no disco, através do uso dos sistemas de memória virtual. Na realidade todos os módulos do sistema têm uma intera-



ção contínua com o disco, mas sempre através dos meios disponíveis, como seja, a função específica denominada SAPO ou os sistemas de memória virtual.

Os módulos de análise recolhem a informação necessária na base de dados criada pela linguagem orientada e a organizam segundo sua função. Até agora se tem implementado módulos de análise linear, análise não linear, e cálculo de autovalores e autovetores. Eles se encarregam de selecionar, segundo seus fins, aquelas rotinas de serviço que se adaptam às exigências do problema.

Na realidade, os módulos de análise são programas iniciados pelo usuário e são totalmente independentes entre si, ainda que possam utilizar rotinas de serviço comuns.

Os módulos de serviço são pacotes de subrotinas que recebem pelos parâmetros, a informação necessária para realizar um trabalho. Este trabalho é geralmente muito específico, e quando cometem algum erro no processo, simplesmente interrompem a execução, transmitindo uma mensagem; no entanto, não possuem nenhuma capacidade de decisão para retomar a análise. Há rotinas de serviço manejadas diretamente por uma linguagem orientada, porque ajudam nas funções de I/O, tais como as rotinas de geração de malhas e a rotina de escritura de dados e resultados. As outras subrotinas servem diretamente aos módulos de análise, e estão em constante interação com eles. Existem rotinas para montagem das matrizes dos elementos na matriz global, rotinas para a construção das matrizes de rigidez por elementos, rotinas para solução de sistemas de equações pelo método de triangulação de Gauss, rotinas que realizam análises por Jacobiano generalizado e por iteração por sub-espacos, rotinas para multiplicar matrizes por vetores, para multiplicar matrizes, para somá-las, para checar convergência, etc. Todas estas rotinas, geralmente encarregadas de realizar algum procedimento numérico, podem utilizar, de acordo com a informação fornecida pelos módulos de análises, qualquer outra subrotina do mesmo nível. Assim por exemplo, a rotina que realiza o Newton Raphson deve utilizar, dentre outras, a rotina de resolução de sistemas.

Para que o sistema seja verdadeiramente versátil, as subrotinas de serviço devem ter uma compatibilidade absoluta com qualquer módulo de análise que as queira utilizar. Para isso devem ser programadas de forma muito geral, tratando de cobrir uma grande variedade de casos, com poucos parâmetros. Deve-se ter extremo cuidado em que a forma de armazenar a matriz de rigidez, por exemplo, seja perfeitamente conhecida e adequada para a utilização de todas as rotinas relacionadas com seu manejo.

Todo o funcionamento do sistema, e também o que o faz viável para sua implementação em minicomputadores é o uso do disco como memória de suporte. Na verdade, todos os arranjos de tamanho grande, contendo conectividades, coordenadas dos nós, nomes dos nós, cargas, condições de contorno, etc., se encontram armazenados no disco. Assim como, as matrizes de rigidez e de massa, no caso em que haja, e os vetores contendo a resposta da análise.

3.3 - COMUNICAÇÃO COM O DISCO

Dispõe-se, realmente, de duas formas de comunicação com o disco. A primeira consiste em um conjunto de rotinas capazes de manejar um grande número de arranjos armazenados em disco, o que faz simulando um sistema de memória virtual que funciona de forma similar ao dos sistemas implementados em máquinas maiores: quer dizer, divide o espaço disponível na memória primária em um número adequado de "páginas" ou blocos de informação que são atualizados e levados ao disco para trazer novos blocos necessários à fase particular do programa que se está executando. Para isso escolhe o bloco que foi menos "tocado" ou utilizado nas últimas operações, e este é o que copia no disco para por o requerido no seu lugar. Este sistema de memória virtual foi implementado na Universidade Central da Venezuelapara que servisse às necessidades de qualquer programa a ser realizado no IMME (Instituto de Materiales y Modelos Estructurales) e seu funcionamento foi explicado nas referências [12], [13] e [14].

Este sistema tem a vantagem de ser geral e utilizável por qualquer dos módulos já programados. É requerido, somente, que se declare aqueles arranjos que vão estar armazenados em disco, dando suas características necessárias: dimensões aproximadas, tipo de variável que contém, etc. Posteriormente, quando se requer determinada posição de um arranjo, esta é pedida através de uma subrotina, identificando o arranjo com um número, e o sistema se encarrega de colocá-la em um "common" em memória direta, devolvendo a direção do "common" onde a dita posição se encontra. Além disso, o sistema de memória virtual é capaz de eliminar aqueles arranjos que já não são necessários, compactando o disco, criar arquivos permanentes ou não, etc. Foi comprovada, reiteradamente, sua compatibilidade e a facilidade para ser adaptado a qualquer programa.

[14]

No entanto, se dispõe de outra forma de comunicação com o disco, cuja capacidade não é tão ampla, mas muito mais específica, porém apresentando a vantagem de ser muito mais rápida que a anterior, em algumas aplicações. Tal é o caso da função SAPO, que será descrita a seguir.

A função SAPO armazena um único arranjo no disco, razão porque, todos os trabalhos de identificação são minimizados. Na verdade, esta função foi especialmente concebida para ser utilizada com rotinas de resolução de sistemas de equações por técnicas de eliminação [11], razão pela qual leva implícito em seu funcionamento o conceito de "FILA PIVOTE", isto é, a fila que num momento determinado está sendo utilizada para 'eliminar' outras, é marcada por um sinal para obrigar sua permanência em memória direta.

Neste momento, todos os módulos de análises que se apresentam neste trabalho foram implementados através da função SAPO, no que diz respeito ao manejo da matriz de rigidez, matriz de massa e vetores de solução. Todavia, tudo que se refere à utilização de dados e saída de resultados, (uso da linguagem orientada) foi resolvido com sistema de memória virtual, já mencionado, que, envolvendo o manejo de vários arran

jos, parece mais adequado para estas situações. Por outro lado, os processos que envolvem mais tempo de máquina, são os de análises, razão pela qual é conveniente a utilização, nesta etapa, de algoritmos mais simples e menos custosos.

Para um algoritmo de solução de equações devem encontrar-se em memória direta pelo menos duas filas da matriz, por tanto decidiu-se definir um "common" tão grande quanto possível e dividido em dois blocos de igual tamanho. O disco conteria uma fileira deste blocos, identificados por números, em quantidade suficiente para conter todo o arranjo definido. (ver Fig. 3.2).

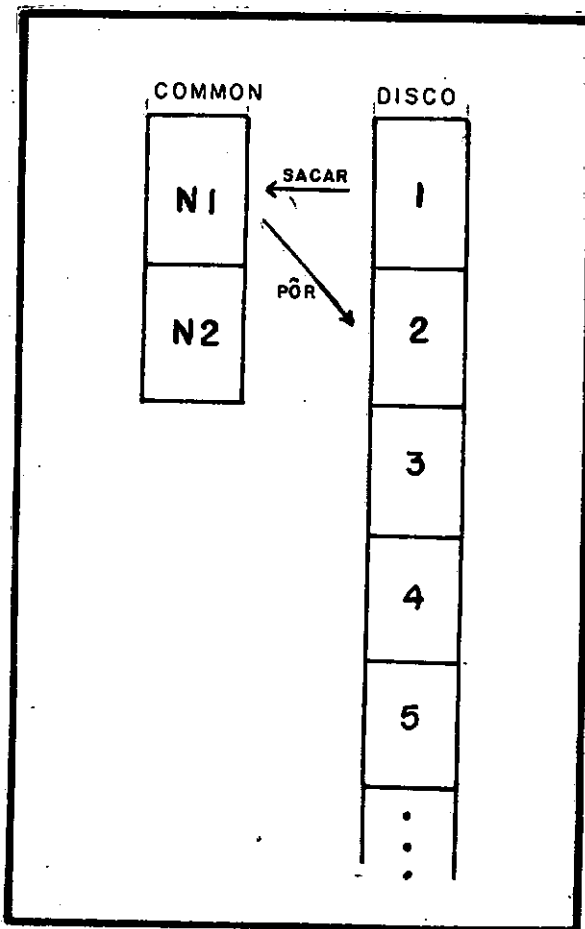


Figura 3.2
A Função SAPO

A matriz se encontra armazenada em filas, uma atrás da outra, no disco. Como se conhece o número de filas que constituem um dos blocos, pode-se determinar facilmente a que bloco pertence uma determinada fila.

Em consequência, para a fila I, se NFS é o número de filas por bloco, corresponderá ao bloco N° IS:

$$IS = \frac{I - 1}{NFS} + 1 \quad (3.1)$$

O seguinte passo que realiza a função é averiguar se o bloco IS se encontra em memória direta. Se não está, identifica qual é o bloco que contém a fila pivote, e que deve permanecer, e coloca o outro em sua posição correspondente no disco. Posteriormente, copia o bloco IS no segmento disponível do "common" em memória direta e devolve a posição do carácter (I,J) da fila I, que foi requerido.

Consequentemente, dentro de determinado bloco, se MS é a largura de uma fila, a posição de (I,J) será:

$$IPOS = (I - (IS - 1) * NFS - 1) * MS + J \quad (3.2)$$

Os parâmetros da função SAPO incluem um código que permite utilizá-la de três maneiras:

- Extrai um elemento (I,J) da matriz.
- Coloca uma quantidade dada na posição (I,J) da matriz.
- Soma uma quantidade dada à posição (I,J) da matriz.

A função SAPO devolve tanto o valor do carácter(I, J) como sua posição dentro do "common" em memória direta.

Isto permite trabalhar por caracteres ou por filas. Por exemplo, Se está se resolvendo um sistema de equações e se traz o primeiro carácter de uma fila à memória direta, se sabe

NÚMERO DE EQUAÇÕES	SEMI BANDA	NÚMERO TOTAL DE ELEMENTOS NA MATRIZ	DIMENSÃO DO COMMON	TEMPO TEMPO (seg)
25	7	175	1450	2.00
81	10	810	2500 1450 600 200	2.32 2.32 2.49 3.30
85	12	1020	2500 1450 600 200	3.26 3.26 3.55 9.74
108	20	2160	2500 1450 600 200	4.30 6.73 9.48 24.48
114	13	1482	2500 1450 600 200	4.03 12.76 17.26 44.94
176	20	3520	1450 600 200	22.60 48.32 95.96
400	84	33600	1450 600 200	237.5 614.6 1356.0

Tabêla 3.1
Tempos Obtidos para a Solução Utilizando "SAPO

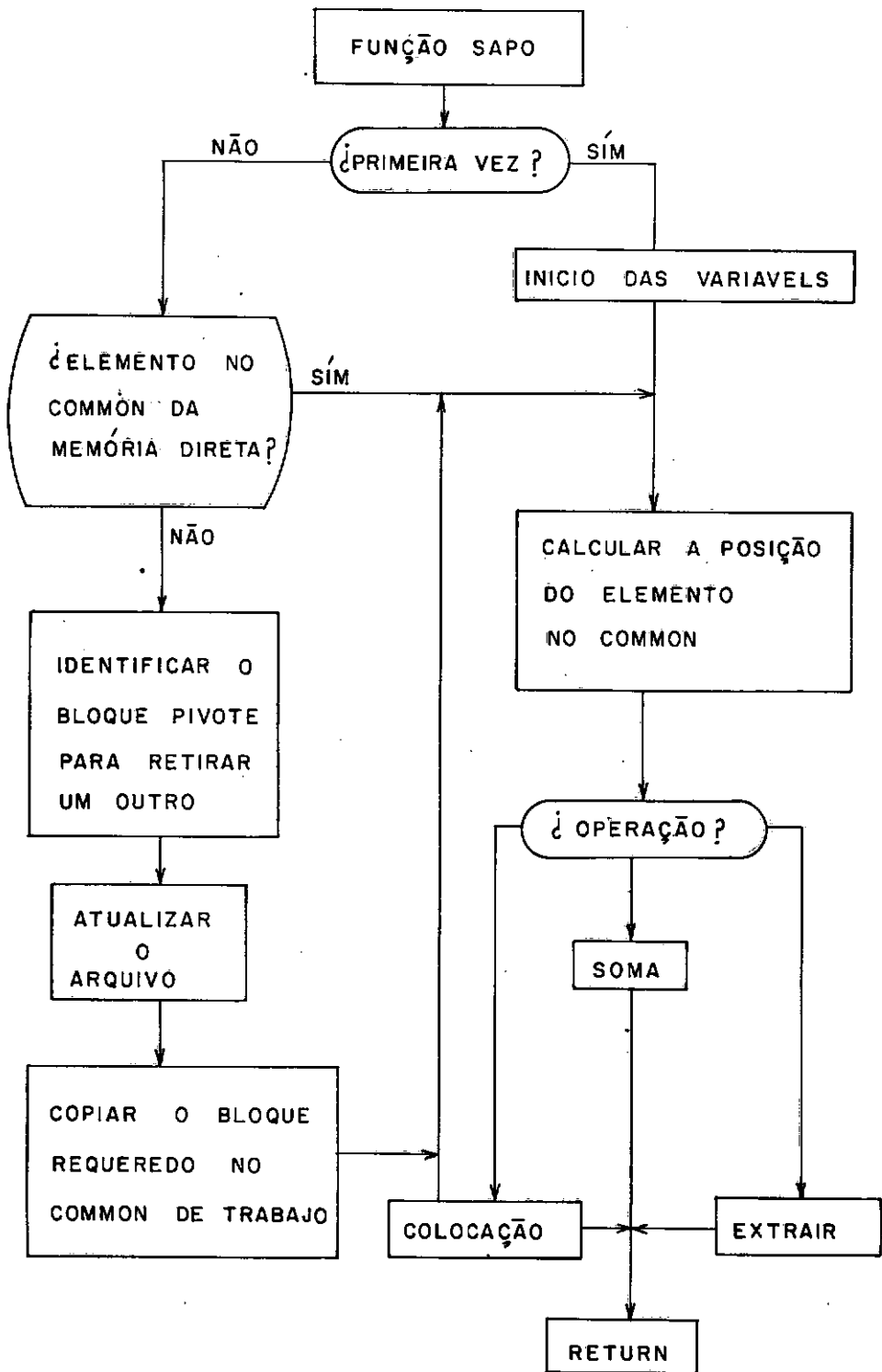


Fig 3.3
Função Sapo

que toda a fila foi traída, pelo que, pode-se aproveitar esta característica para evitar chamadas desnecessárias à Função SAPO.

A Fig. 3.3 mostra um diagrama simplificado do funcionamento de "SAPO" [28].

Esta função é eficiente, no sentido de ser utilizável nos casos descritos com um tempo razoável. A Tabela 3.1 indica alguns tempos obtidos para resolução de sistemas de equações de diferentes tamanhos.

Ainda que a função SAPO maneje o disco como um único arranjo, em casos como a análise por iteração por sub-espacos foram armazenados além da matriz de rigidez, a matriz de massa, os vetores de iteração e vetores auxiliares. Simplesmente, os vetores foram armazenados com se estivessem formando parte das filas da matriz de rigidez (aumentando a largura de banda) e a matriz de massa continuando a matriz de rigidez.

3.4 - EXPANSÕES PREVISTAS

Seguindo a mesma filosofia que mostra a arquitetura do sistema na Fig. 3.1, tem-se expansões previstas, tanto dos módulos de análise que cobrem o sistema, como das rotinas de serviço disponíveis. Assim sendo, se desenvolverá a linguagem orientada mediante a adição de comandos mais poderosos e gerais.

Na realidade, este trabalho cumpre uma das metas da primeira etapa de um sistema que está sendo desenvolvido por um grupo de Professores do IMME (Instituto de Materiales y Modelos Estructurais) da Universidad Central de Venezuela [36]. Pretende-se que uma versão mais avançada deste sistema cubra as necessidades (pelo menos quanto a modelos matemáticos e numéricos) que surgem em diversas áreas na hora de empreender qualquer projeto de pesquisa em Engenharia Civil. Neste sentido, o IMME sempre precisou de um sistema de computador próprio e adaptado às necessidades do Instituto. Por outro lado, estando este insti-

tuto tão ligado a tarefas docentes, por depender da Faculdade de Engenharia, a utilização do sistema se estende ainda mais e passa do âmbito da pesquisa no âmbito da educação, podendo -se aproveitar para o ensino de matérias básicas do Engenheiro e como um estímulo para a realização de teses de graduação, trabalhos extras, etc. Em tal sentido, é evidente a necessidade de desenvolver uma linguagem orientada mais poderosa que a existente e o suficientemente generalizada, para poder cobrir a grande variedade de necessidades que se desejam suprir.

A arquitetura modular do sistema facilita em alto grau a resolução de novos problemas de aplicação. Em consequência, definido o problema, se junta o comando da linguagem orientada correspondente, assim como também todas aquelas rotinas de serviço necessárias, que não se encontram já dentro do sistema. Para isso devem se utilizar as formas de comunicação com o disco, e garantir ademais, uma perfeita compatibilidade destas rotinas com qualquer dos módulos já existentes. Desta forma, as novas rotinas serão facilmente utilizadas para a programação de outros módulos de análises.

Dentre os módulos de análises que se projeta implementar em um futuro próximo, está o correspondente à análise dinâmica, abrangendo vibrações forçadas, com todos os módulos de serviço que isto requerirá (algum método de integração direta, etc.)

A nível de módulos de serviço, planeja-se criar uma verdadeira biblioteca de elementos, triangulares, quadriláteros, etc., não só para aplicações da equação de campo generalizada, como para aplicações em elasticidade bidimensional, problemas de estruturas de barra, pórticos, etc. Além disso, outras formas de trabalhar com a matriz de rigidez, tais como: armazenamento por "SKYLINE" [27], resolução por submatrizes, pelo método frontal, por Cholesky, etc. Isto permitirá oferecer ao usuário um número adequado de opções para que possa escolher a mais adequada ao seu problema. Entre outras, também serão implementadas rotinas para condensação, o que permitirá realizar análises por sub-estruturas; além disso, rotinas de geração de

malhas considerando a variedade de elementos em biblioteca, rotinas de saída gráfica, incluindo desenhos no plotter e no printer, etc.

Quando a diversidade de rotinas de pre-processamento, análises e post-processamento o requeram, se fará necessária a implementação de um programa executivo que coordene o funcionamento do sistema e efetue a interfase entre rotinas; isto é conveniente para dar mais amplitude e flexibilidade ao sistema. Atualmente os trabalhos do executivo são feitos por cada um dos módulos de análises mostrados na Fig.(3.1).

Concluindo, foram descritas as bases do que será um sistema de elementos finitos destinado a problemas de engenharia civil, e especialmente adaptado a minicomputadores, por quanto resolve o problema da limitação da memória nestas máquinas. Neste trabalho se descreve o esquema mostrado na Fig. 3.1, que representa grande parte do já existente. Outros módulos já em funcionamento, ainda que não completamente integrados ao sistema, são os de aplicação a problemas de elasticidade plana, a problemas de pórticos e de barras, mas somente no que se refere à sua análise linear.

CAPÍTULO IV

APLICAÇÃO: A EQUAÇÃO DE
CAMPO GENERALIZADA

4.1 - INTRODUÇÃO

Os módulos em funcionamento do sistema, mostrados na Fig. 3.1 foram aplicados neste trabalho a problemas regidos pela equação de campo generalizada em seus casos estáveis, isto é, a problemas independentes do tempo.

Existem numerosos problemas de importância prática que são regidos por este tipo de equação. Tal é o caso de problemas associados com torsão de barras prismáticas, fluxo através de meios porosos, condução de calor, geração interna de calor, condução elétrica, lubrificação, vibrações acústicas, transmissão de ondas magnéticas, etc.

Neste capítulo são brevemente mencionados e explicados, alguns dos problemas cobertos por esta equação, e o significado dos parâmetros que envolve para cada caso. Posteriormente, no Capítulo VII se exemplificam alguns casos, com seus resultados e comparações.

Adicionalmente, apresenta-se neste capítulo uma formulação discreta para problemas bidimensionais, com a formulação matricial que envolve o uso de elementos finitos triangulares simples. Esta formulação é a que se encontra implementada para os módulos em funcionamento.

Deve-se insistir em que a equação de campo generalizada cobre somente uma parte das aplicações do sistema, já que os módulos de análise e de serviço disponíveis (Fig. 3.1) são totalmente gerais e podem ser utilizados para qualquer outro tipo de problema de aplicação.

4.2 - A EQUAÇÃO DE CAMPO GENERALIZADA

A forma mais geral desta equação corresponde à fórmula (4.1)

$$\frac{\partial}{\partial x} \left(h_x \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(h_y \frac{\partial \phi}{\partial y} \right) + \frac{\partial}{\partial z} \left(h_z \frac{\partial \phi}{\partial z} \right) + \lambda \phi = c + \rho \frac{\partial T}{\partial t} \quad (4.1)$$

Neste capítulo, no entanto, só se tratou de problemas bidimensionais e independentes do tempo, razão pela qual a fórmula fica reduzida a:

$$\frac{\partial}{\partial x} \left(h_x \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(h_y \frac{\partial \phi}{\partial y} \right) + \lambda \phi = C \quad (4.2)$$

donde ϕ é uma quantidade potencial, cujo valor varia com a posição, λ é uma constante, C pode variar em função das coordenadas e h_x e h_y são características físicas, também variáveis.

Em geral, a expressão (4.1) não é senão o estabelecimento do princípio de continuidade para a função de transferência de um fluxo determinado q , tal que: [18]

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} = C \quad (4.3)$$

$$e: \quad q_x = K_x \frac{\partial \phi}{\partial x}$$

$$q_y = K_y \frac{\partial \phi}{\partial y}$$

$$q_z = - K_z \frac{\partial \phi}{\partial z} \quad (4.4)$$

As condições de bordo que acompanham as equações (4.1) e (4.2) podem ser de dois tipos: [35]

a) De Dirichlet, da forma

$$\phi = \bar{\phi} \quad (4.5)$$

onde ϕ é um valor conhecido.

b) De Cauchy, da forma

$$K_x \frac{\partial \phi}{\partial x} n_x + K_y \frac{\partial \phi}{\partial y} n_y = F(x,y) + \phi h(x,y) \dots \quad (4-6)$$

onde n_x e n_y são cossenos diretores da normal n .

É conveniente assinalar que em um ponto dado do bordo só pode ocorrer uma das duas condições (4.5) ou (4.6) ao mesmo tempo.

As equações mencionadas (4.2), (4.5) e (4.6) podem ser aplicadas a uma grande quantidade de problemas físicos, alguns dos quais foram mencionados no ponto (4.1), basta somente se conhecer adequadamente que significam as variáveis utilizadas para cada problema de aplicação. A tabela (4.1) proporciona um resumo dos problemas com o significado físico das variáveis.

22

Os casos particulares da equação (4.2) são chamados:

- Equação de Poisson:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \frac{C}{K} \quad (4.7)$$

- Equação de Laplace:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (4.8)$$

- Equação de Hemboltz:

$$\frac{\partial}{\partial x} \left(K_x \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y \frac{\partial \phi}{\partial y} \right) + \lambda \phi = 0 \quad (4.9)$$

Problema		K_x, K_y	C	$f(x, y)$	$h(x, y)$
Fluxo em meios porosos	Altura de Pressão	Permeabilidade do meio	Fonte interna de fluxo	Fluxo de Contorno	-
Condução elétrica	Voltagem	Condutividade elétrica	Fonte interna de corrente	Corrente aplicada a um contorno	-
Campo eletro-estático	Intensidade de campo elétrico	Permissividade	Fonte interna de corrente	-	-
Lubrificação	Pressão	Funções de espessor e viscosidade	Fluxo direto devido a várias ações	Fluxo de contorno	-
Gravitação	Componente da força gravitacional por unidade de massa	-	-	-	-
Transmissão de calor	Temperatura	Condutividade térmica	Geração de calor interno	Fluxo de calor no contorno	Coefficiente de convecção
Fluxo irrotacional	Potencial de velocidade ou função corrente	1	-	Velocidade no contorno	-
Torsão de barras prismáticas	Função esforço	Inverso do módulo de corte $\frac{1}{G}$	Ângulo de giro por unidade de comprimento	-	-

Infiltração lenta ("seepage")	Pressão	Permeabi- lidade	Fonte de fluxo in- terna	-	-
Magnetismo	Força magnética	Permeabi- lidade magnética	Fonte in- terna de campo mag- nético	Intensida- de de cam- po aplica- da ao bor- do.	

Tabela 4.1
 Identificação das variáveis
 para a equação de campo

4.3 - EXEMPLOS PARTICULARES DE APLICAÇÃO

Antes de entrar com a formulação para aplicar o método dos elementos finitos à equação (4.1), faz-se nesta seção uma descrição mais detalhada de alguns exemplos de aplicação mencionados, os quais podem ser resolvidos com o sistema descrito.

4.3.1 - Transferência de Calor

Este é um processo pelo qual a energia se transfere das regiões de mais alta temperatura às de mais baixa temperatura.

Existem três mecanismos diferentes de transmissão de calor: 29

- condução
- convecção
- radiação.

Dos três, a radiação não será estudada, e a transferência por convecção aparecerá como uma condição de contorno nos exemplos considerados.

A taxa em que o calor é transmitido por condução se calcula:

$$q_x = K_x \cdot A \frac{dT}{dx} \quad (4.10)$$

donde K_x é a condutividade térmica, A a área da seção através da qual se transfere o calor, x é a direção do fluxo de calor e $\frac{dT}{dx}$ o gradiente de temperatura na direção x .

A tabela (4.2) mostra valores típicos da condutividade térmica para alguns materiais.

MATERIAL	CONDUTIVIDADE TÉRMICA a 27°C ($\frac{\text{Watts}}{\text{m}^\circ\text{K}}$)
Cobre	386
Alumínio	204
Aço Carbônico	54
Vidro	0.75
Plásticos	0.2 - 0.3
Água	0.6
Óleo Combustível	0.15
Hidrogênio	0.18
Ar	0.026

Tabela 4.2

Valores de K_x e K_y

Para muitos materiais a condutividade térmica não é uniforme, mas varia com a temperatura. Na maioria dos casos, é possível aproximar essa variação como função linear, se a temperatura se encontra dentro de certas faixas:

$$K(t) = K_0 (1 + \beta T) \quad (4.11)$$

onde K_0 é o valor da condutividade térmica para uma temperatura

ra de referência e β é uma constante determinada empiricamente.

Para um caso de transmissão de calor em uma só direção como o que mostra a Figura (4.1), a distribuição de temperatura vem dada pela fórmula adimensional: (4.12)

$$\frac{T(x)-T_1}{T_2-T_1} = \frac{x}{L} + \frac{\beta}{2} \left[(T_2-T_1) \frac{x}{L} + \frac{T_1 - T(x)^2}{T_2-T_1} \right] \quad (4.12)$$

de forma que quando $\beta = 0$, transforma-se no caso de K constante:

$$T(x) = \frac{x}{L} (T_2 - T_1) + T_1 \quad (4.13)$$

onde T_2 é a temperatura para $x = L$, e T_1 a temperatura para $x=0$.

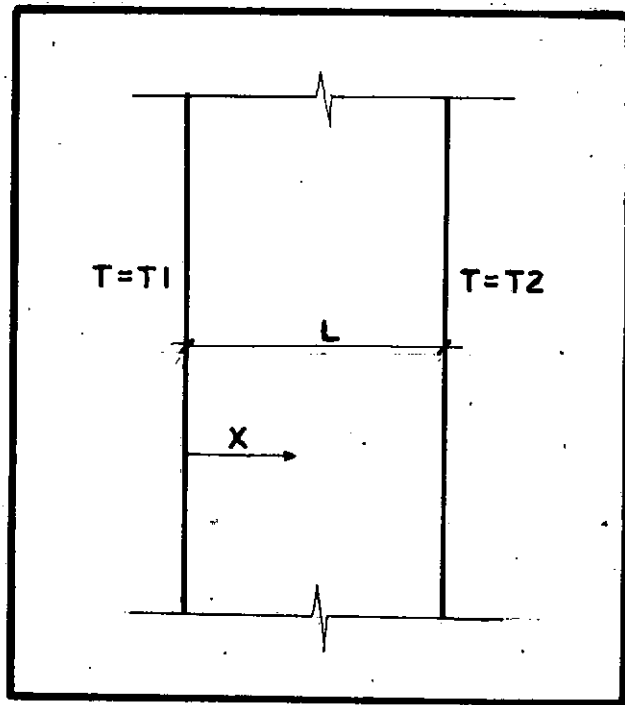


Figura 4.1

As condições de convecção se produzem quando um fluido entra em contacto com uma superfície sólida que se encontra numa temperatura diferente [29] e [32]. Pode haver convecção "livre" ou convecção "forçada" (Fig. 4.3) mas em ambos os casos a transmissão de calor na interfase se realiza por CONDUÇÃO, e a taxa de calor (q_c) transmitido se escreve pela seguinte Fórmula ("Lei de Newton do esfriamento").

$$q_c = - KF \left. \frac{T}{y} \right|_{y=0} = \bar{h}_c . A (T_s - T_\alpha) \quad (4.14)$$

onde \bar{h}_c é a condutividade térmica para convecção (ver Tabela 4.3), A é a área da superfície através da qual se transmite o fluxo, T_s é a temperatura na superfície e T_α a temperatura do fluido distante da superfície de transferência.

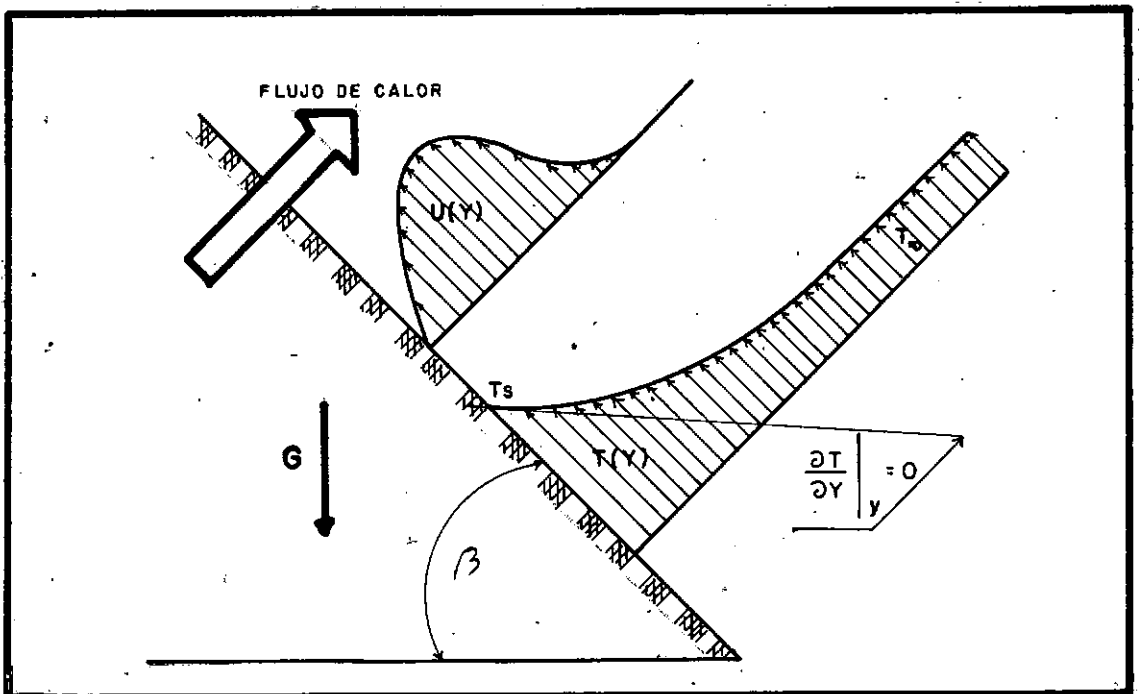


Figura 4.2

CONVECÇÃO LIVRE: O movimento se produz pelas diferenças de densidade no fluido, produto de um contacto com uma superfície quente.

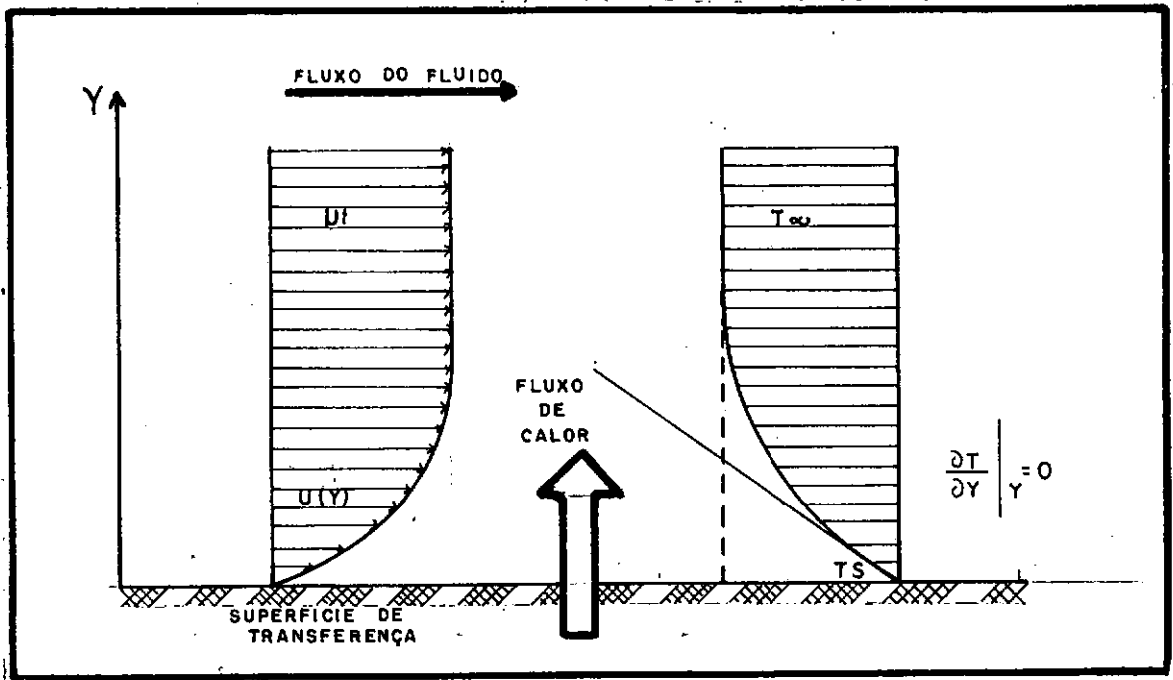


Figura 4.3

CONVECÇÃO FORÇADA: ocorre quando um fluido à velocidade dada U_F passa por uma superfície com temperatura diferente.

FLUIDO E TIPO DE CONVECÇÃO	$\bar{h}_c \left(\frac{\text{W}}{\text{m}^2 \text{OK}} \right)$
Convecção Livre, Ar	5 - 25
Convecção Livre, Água	20 - 100
Convecção Forçada, Ar	10 - 200
Convecção Forçada, Água	50 - 10000
Água Fervente	3000 - 1000000
Condensação da Água	5000 - 1000000

Tabela 4.3

Condutividade Térmica
para Convecção

Concluindo, os problemas de transferência de calor que enfrenta o sistema, são regidos pela equação (4.2) onde λ é nulo e C é a taxa de calor interno gerada dentro do recinto. As condições de contorno vêm expressadas por (4.5) (Dirichlet), (4.10) (fluxo de condução) e (4.14) (fluxo de convecção). Observa-se não linearidade segundo a equação (4.11).

4.3.4 - Problemas de Torção:

Torção de Saint Venant

Segundo a formulação de Saint-Venant [33] o problema de torção de barras prismáticas é representado de acordo com a Figura 4.4.

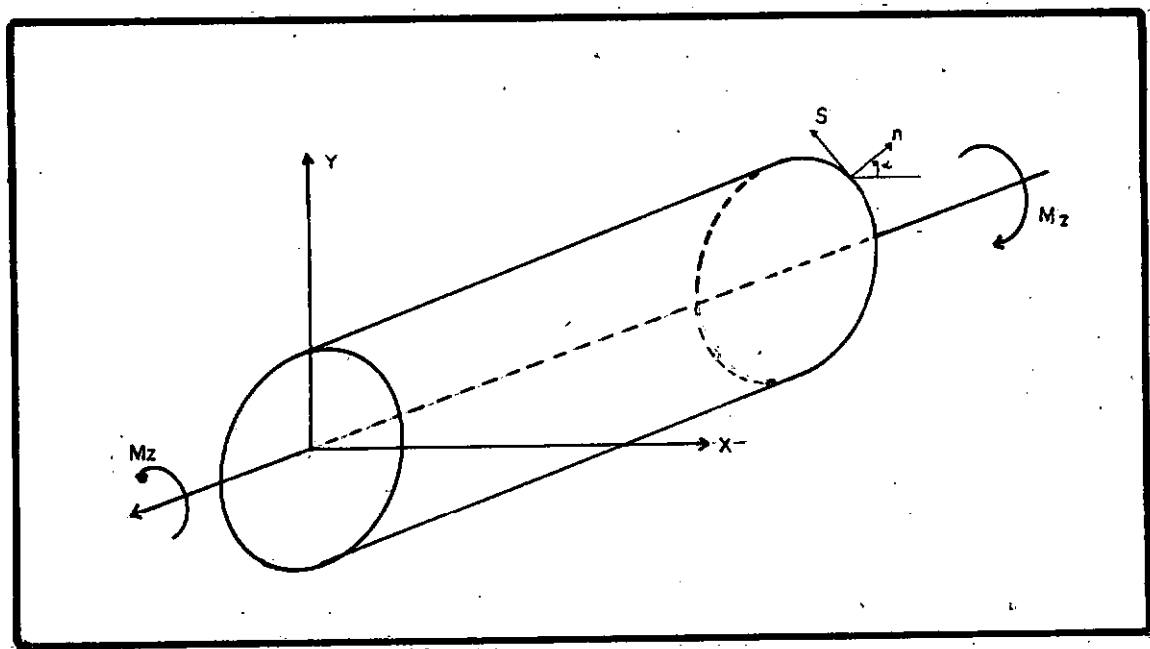


Figura 4.4
Torção de Saint-Venant

Todas as seções se deformam da mesma maneira [23], e a equação que rege o fenômeno é a equação de Laplace.

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \quad (4.15)$$

onde ψ é a função de torção.

Os deslocamentos vêm dados por:

$$\begin{aligned} u &= -\theta_{zy} \\ v &= \theta_{zx} \\ w &= \theta\psi(x,y) \end{aligned} \quad (4.16)$$

onde θ é o ângulo de rotação por unidade de longitude.

Os esforços vêm dados por:

$$\begin{aligned} \sigma_x = \sigma_y = \sigma_z = \tau_{xy} &= 0 \\ \tau_{zx} &= G\theta \left(\frac{\partial \psi}{\partial x} - y \right) \\ \tau_{yz} &= G\theta \left(\frac{\partial \psi}{\partial y} + x \right) \end{aligned} \quad (4.17)$$

onde $G = \frac{E}{2(1+\nu)}$

Outra forma de modelar o problema é através da teoria de Prandtl a qual define a função esforço ϕ , onde:

$$\frac{1}{G} \frac{\partial^2 \phi}{\partial^2} + \frac{1}{G} \frac{\partial^2 \phi}{\partial^2} = 2\theta \quad (4.18)$$

que é uma equação de Poisson.

Os esforços se obtêm de :

$$\begin{aligned} \tau_{xz} &= \frac{\partial \phi}{\partial y} \\ \tau_{yz} &= -\frac{\partial \phi}{\partial x} \end{aligned} \quad (4.19)$$

4.3.3 - Problemas de Fluxo Potencial:

Considera-se o fluxo em movimento irrotacional, não se levam em conta as forças de atrito e o fluido é incompressível (cumpre-se o princípio de continuidade). [28]

Estes problemas podem ser resolvidos de duas formas diferentes:

- Com base na função de corrente Ψ
- Com base no potencial ϕ

Foram feitas comparações destes problemas com os de torsão 33

As velocidades se definem por:

$$\begin{aligned} V_x &= \frac{\partial \phi}{\partial x} = \frac{\partial \Psi}{\partial y} \\ V_y &= \frac{\partial \phi}{\partial y} = - \frac{\partial \Psi}{\partial x} \end{aligned} \quad (4.20)$$

Ambas as funções completam a equação de Laplace no seio do fluido.

4.3.4 - Problemas Envolvendo a Equação de Hemholtz

Estes problemas geralmente se referem à propagação de diferentes classes de ondas. Tal é o caso das ondas eletromagnéticas, ondas acústicas e oscilação das ondas em um corpo plano de água ("seiche motion"), entre outros. [22]

A equação de Hemholtz tem a forma:

$$\frac{\partial}{\partial x} \left(K_x \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y \frac{\partial \phi}{\partial y} \right) + \lambda \phi = 0 \quad (4.21)$$

com as condições de contorno:

$$\phi = \bar{\phi} \quad (4.22)$$

$$K_x \frac{\partial \phi}{\partial x} n_x + K_y \frac{\partial \phi}{\partial y} n_y = 0 \quad (4.23)$$

A constante λ pode ou não ser conhecida; neste último caso, a formulação da equação (4.21) leva à resolução de um problema de autovalores e autovetores.

VIBRAÇÕES ACÚSTICAS

Um fluido que vibra dentro de um corpo fechado produz um campo de ondas esféricas, que pode ser representado pela equação (4.)

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\omega^2}{c^2} p = 0 \quad (4.24)$$

onde P é o excesso de pressão sobre a pressão ambiental ω é a frequência da onda no meio. 22

A equação (4.23) se baseia no fato de que o processo é adiabático, as mudanças locais na densidade são pequenas, assim como os deslocamentos e a velocidade do fluido.

ONDAS ELETROMAGNÉTICAS

A propagação destas ondas em um recinto cheio de material dielétrico se rege pela equação: [22]

$$\frac{\partial}{\partial x} \left(\frac{1}{\epsilon_d} \frac{\partial \phi}{\partial x} \right) + \frac{1}{\partial y} \left(\frac{\partial \eta}{\epsilon_d} \frac{\partial \phi}{\partial y} \right) + \omega^2 \Gamma_0 \epsilon_0 \phi = 0 \quad (4.25)$$

onde ϕ é uma componente do campo elétrico ou magnético, ω é a frequência da onda, Γ_0 é a permeabilidade do espaço livre, ϵ_s a permissibilidade do espaço livre, e ϵ_d a do dielétrico.

OSCILAÇÕES EM PORTOS E BAIAS 22 23 34

Para equações lineares de ondas de longo período, assume-se, geralmente, que a fricção pode ser desprezada, os termos de aceleração convectiva são pequenos e as velocidades podem ser consideradas constantes desde a superfície da água até o fundo. Caso se possam assumir essas considerações, a equação que governa as vibrações livres de um corpo de água é:

$$\frac{\partial}{\partial x} \left(h \frac{\partial \eta}{\partial x} \right) + \frac{\partial}{\partial y} \left(h \frac{\partial \eta}{\partial y} \right) + \frac{4\pi^2}{gT^2} \eta = 0 \quad (4.26)$$

onde h é a profundidade, η é a elevação da água em relação ao nível médio de águas tranquilas e T é o período de oscilação.

Pode-se utilizar o método dos elementos finitos para se encontrar os períodos naturais e os modos de vibração de portos que possam ser aproximados como corpos completamente fechados de água. [23] A solução exata nestes casos é dada para os diferentes modos de vibração em um recinto retangular, pelos termos de uma série:

$$\eta = \cos \frac{m\pi x}{a} \cos \frac{n\pi y}{b} \quad (4.27)$$

onde a e b são as dimensões do corpo fechado (a é o comprimento e b a largura).

Mesmo assim os períodos podem ser calculados por. [34]

$$T_{mn} = \frac{2a}{\sqrt{gd} (m^2 + \beta^2 n^2)^{1/2}} ; \quad \beta = \frac{a}{b} \quad (4.28)$$

onde d é a profundidade da água.

Explicações mais detalhadas de outros problemas propostos na Tabela 4.1 se encontram nas referências [18], [23], [28], [25] e [24]

4.4 - FORMULAÇÃO DISCRETA PARA PROBLEMAS BASEADOS NA EQUAÇÃO DE CAMPO

Utilizando-se o método dos elementos finitos, aproxima-se a função ϕ , solução exata da equação (4.1) por uma solução aproximada ϕ_1 , que determina certos erros nas equações.

Assim:

$$\frac{\partial}{\partial x} \left(K_x \frac{\partial \phi_1}{\partial x} + \frac{\partial}{\partial y} \left(K_y \frac{\partial \phi_1}{\partial y} \right) + \lambda \phi_1 - C \right) = \epsilon_1 \neq 0 \quad (4.29)$$

$$K_x \frac{\partial \phi_1}{\partial x} n_x + K_y \frac{\partial \phi_1}{\partial y} n_y - f - h_1 = \varepsilon_2 \neq 0 \quad (4.30)$$

onde ε_1 e ε_2 são erros de aproximação [35].

Para se realizar a formulação, supôs-se que a função ϕ_1 satisfizesse a condição de Dirichlet (4.5) com exatidão, e se aplicou o método de Galerkin [18] [24] e [35], para minimizar os erros ε_1 e ε_2 . A expressão de resíduos ponderados que resultou foi:

$$\begin{aligned} \iint_A \left[\frac{\partial}{\partial x} \left(K_x \frac{\partial \phi_1}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y \frac{\partial \phi_1}{\partial y} + \lambda \phi_1 - C \right) \right] \delta \phi_1 \, dA + \\ + \iint_S \left[F + h \phi_1 - K_x \frac{\partial \phi_1}{\partial x} n_x - K_y \frac{\partial \phi_1}{\partial y} n_y \right] \delta \phi_1 \, ds = 0 \end{aligned} \quad (4.31)$$

onde A é o domínio de integração na área e S é o bordo onde existem as condições de contorno (4.).

Integrando-se por partes e operando-se a expressão (4.), resulta:

$$\begin{aligned} A \cdot \left[K_x \frac{\partial \phi_1}{\partial x} \frac{\delta \phi_1}{\partial x} + K_y \frac{\partial \phi_1}{\partial y} \frac{\delta \phi_1}{\partial y} - \lambda \phi_1 \delta \phi_1 + C \delta \phi_1 \right] dA - \\ - \int_S \left[F + h \phi_1 \right] \delta \phi_1 \, ds = 0 \end{aligned} \quad (4.32)$$

A função ϕ_1 , seguindo o método dos elementos finitos, é aproximada por certas funções "N" de interpolação, com base nas incógnitas nodais por elemento, " ϕ_e ", tal como se mostra na seguinte equação:

$$\phi_1 = N \phi_e \quad (4.33)$$

Introduzindo-se esta expressão na equação (4.3), e considerando-se que ϕ_e e $\delta \phi_e$ não são funções de posições, resulta:

$$\delta \underline{\phi}_e^T \left\{ \iint_A (K_x \underline{N}_x^T \underline{N}_x + K_y \underline{N}_y^T \underline{N}_y) dA - \iint_A \lambda \underline{N}^T \underline{N} dA - \right. \\ \left. - \int_S h \underline{N}^T \underline{N} ds \right\} \underline{\phi}_e + \iint_A C \underline{N}^T dA - \int_S F \underline{N}^T ds \} = 0 \quad (4.34)$$

onde as integrações se realizam ao nível do elemento, toda a vez que a expressão (4.3) aproxima a função sobre o elemento em questão.

Considerando, na equação (4.3) que a variação $\delta \underline{\phi}_e$ é arbitrária, o termo à direita deve ser nulo, podendo definir-se então:

$$(\underline{K}^e - \lambda \underline{M}^e - \underline{H}^e) \underline{\phi}^e = \underline{p}^e \quad (4.35)$$

onde o "e" indica que as matrizes estão dadas por elemento, e:

$$\underline{K}^e = \iint_A \left[K_x \underline{N}_x^T \underline{N}_x + K_y \underline{N}_y^T \underline{N}_y \right] dA \quad (4.36)$$

$$\underline{M}^e = \iint_A \underline{N}^T \underline{N} dA \quad (4.37)$$

$$\underline{H}^e = \int_S h \underline{N}^T \underline{N} ds \quad (4.38)$$

$$\underline{p}^e = - \iint_A C \underline{N}^T dA + \int_S F \underline{N}^T ds \quad (4.39)$$

definem as matrizes que se utilizam em (4.3). As vezes, e com algumas semelhanças com problemas tradicionais de estruturas, dá-se o nome de matrizes de "rigidez", de "massa" e de "vetores de carga", respectivamente, às matrizes dadas por (4.36), (4.37) e (4.38).

Estas equações, dadas ao nível do elemento, se reúnem para dar forma ao problema global, que abrange todo o recinto discretizado.

Vale comentar que a equação de Hemholtz se apresenta em problemas onde as funções f, c e h se fazem nulas, o que resulta no problema de autovalores e autovetores:

$$(\underline{K} - \lambda \underline{M}) \underline{\phi} = 0 \quad (4.40)$$

A equação de Laplace seria do tipo:

$$\underline{K} \underline{\phi} = \underline{P} \quad (4.41)$$

Dessa maneira a equação de Poisson ficaria de forma semelhante à anterior.

4.5 - ELEMENTO TRIANGULAR SIMPLES

A aplicação de problemas regidos pela equação de campo se realiza atualmente com base na utilização de elementos triangulares isoparamétricos de 3 nós, que proporcionam uma variação linear da função ϕ sobre o elemento (Fig.4.5)

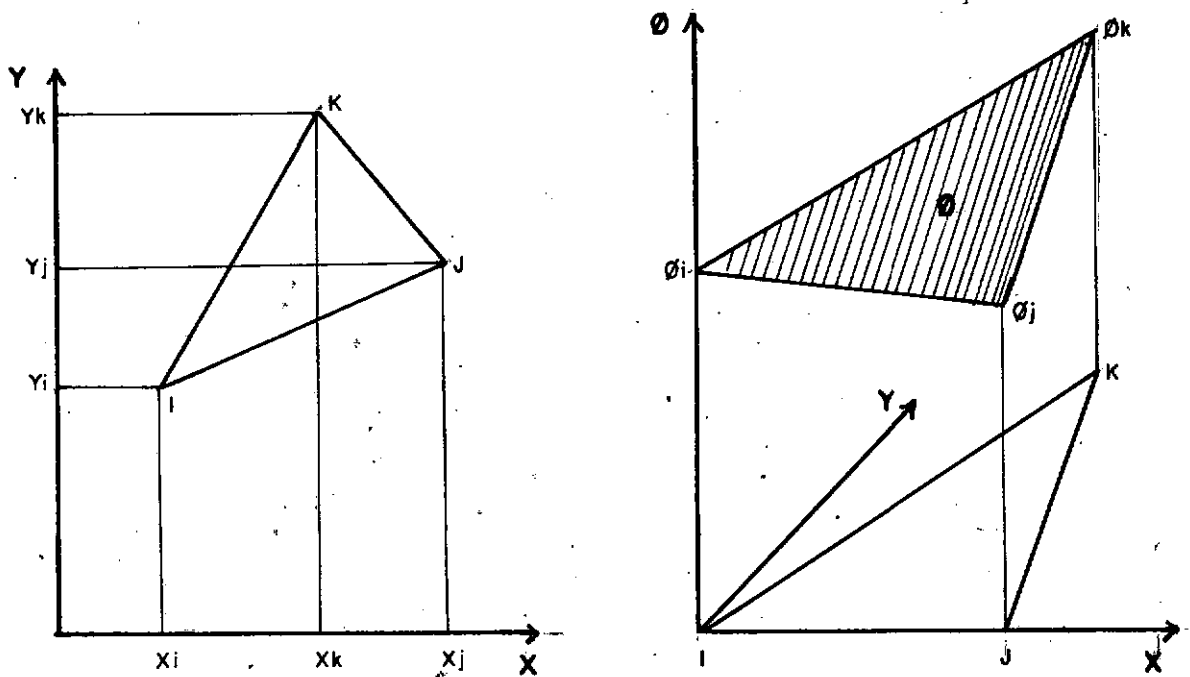


Figura 4.5
Elemento Triangular Simples, Variação
da Função sobre o Recinto

O desenvolvimento explícito das matrizes características deste elemento, desenvolvido com base em funções de interpolação lineares, correspondentes às chamadas "coordenadas triangulares", pode ser encontrado nas referências [35] [25] e [30].

Deste desenvolvimento resultam as matrizes: 35

$$\underline{K}^e = A_e (K_x \begin{vmatrix} C_{12}^2 & C_{12}C_{22} & C_{12}C_{32} \\ & C_{22}^2 & C_{22}C_{32} \\ \text{(SIMÉTRICA)} & & C_{32}^2 \end{vmatrix} + K_y \begin{vmatrix} C_{13}^2 & C_{13}C_{23} & C_{13}C_{33} \\ & C_{23}^2 & C_{23}C_{33} \\ \text{(SIMÉTRICA)} & & C_{33}^2 \end{vmatrix}) \quad (4.42)$$

$$\underline{M}^e = \frac{A_e}{12} \begin{vmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{vmatrix} \quad (4.43)$$

$$\underline{P}^e = \frac{C \cdot A_e}{3} \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\} + \frac{1}{2} \left\{ \begin{bmatrix} F_1 S_1 + f_3 S_3 \\ F_2 S_2 + f_1 S_1 \\ F_3 S_3 + f_3 S_3 \end{bmatrix} \right\} \quad (4.44)$$

$$\underline{H}^e = \frac{1}{6} \begin{vmatrix} 2(h_1 S_1 + h_3 S_3) & h_1 S_1 & h_3 S_3 \\ & 2(h_1 S_1 + h_2 S_2) & h_2 S_2 \\ \text{(SIMÉTRICA)} & & 2(h_2 S_2 + h_3 S_3) \end{vmatrix} \quad (4.45)$$

onde os termos C_{ij} são elementos da matriz \underline{B} (derivadas das fun

ções \tilde{N}):

$$\tilde{B} = \frac{1}{2A_e} \begin{bmatrix} (Y_2 - Y_3) & (X_3 - X_2) & (X_2 Y_3 - Y_2 X_3) \\ (Y_3 - Y_1) & (X_1 - X_3) & (X_3 Y_1 - X_1 Y_3) \\ (Y_1 - Y_2) & (X_2 - X_1) & (X_1 Y_2 - X_2 Y_1) \end{bmatrix} \quad (4.46)$$

Os valores S_i são as longitudes dos lados do triângulo, e os valores h_i e f_i as quantidades (consideradas constantes sobre os lados correspondentes) de convecção e fluxo por condução atribuídos aos bordos onde se hajam definido tais condições de contorno. A nomenclatura utilizada corresponde à Figura (4.6).

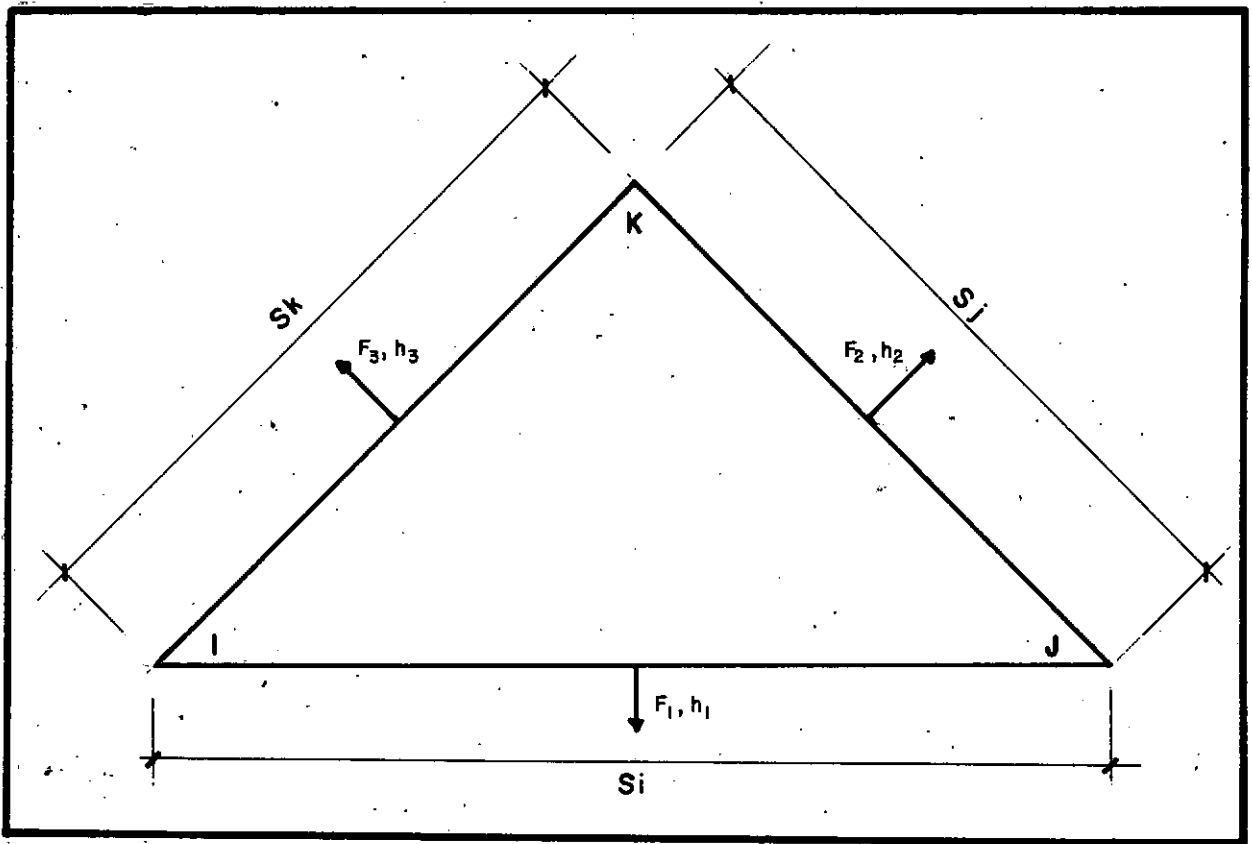


Figura 4.6

Nomenclatura sobre os Lados do Triângulo

CAPÍTULO V

CARACTERÍSTICAS DA LINGUAGEM

5.1 - INTRODUÇÃO

Uma das características que sempre acompanha qualquer conjunto de programas de análises por elementos finitos é a complexidade da entrada de dados. Em consequência, quanto maior a malha utilizada, maior o número de elementos empregados e o número de nós que a formam, com todos os dados inerentes à geometria e propriedades: coordenadas, conectividades, condições de bordo, cargas, etc.

A elaboração da entrada de dados é, pois, um trabalho, que além de ser sumamente tedioso para o usuário, é fonte de inumeráveis e custosos erros, os quais muitas vezes passam despercebidos até mesmo depois da análise.

O conceito de linguagem orientada foi concebido para facilitar esta etapa na resolução de todo o problema. Em consequência, por esse meio se proporciona ao usuário comandos versáteis e poderosos que ajudam na definição do problema com as vantagens de incluir formato livre, vários comandos em um cartão, mensagens de erro, etc.

Deve-se assinalar, como outro aspecto do que representa uma linguagem orientada, a qual constitui numa forma de comunicação familiar e fácil de manejar para o engenheiro, já que se baseia em palavras da terminologia usual empregada em engenharia. Isto facilita a aprendizagem da forma de utilização do sistema e favorece o seu uso em níveis pedagógicos.

A idéia dos "comandos", definidos por palavras chave que identificam diferentes etapas na definição do problema (COORDENADAS, CONECTIVIDADES, etc.) se adaptam muito bem a um sistema modular, já que facilitam a adição de novos esquemas de geração de malha, de condensação, etc., dentro da mesma filosofia e sem

afetar os comandos já existentes.

Para este trabalho, em particular, se busca que a entrada de dados constitua um módulo independente dos módulos de análises, construindo-se uma base de dados que é armazenada em disco para o uso posterior de outros programas.

Na construção deste arquivo permanente se utilizam as facilidades brindadas por um conjunto programado de rotinas, que simulam memória virtual [12]. Isto permite evidenciar a série de dificuldades impostas pela limitação de memória no minicomputador, dispondo-se assim de capacidade suficiente para resolver problemas de dimensão respeitável.

Tanto os programas que constituem a linguagem orientada como o módulo que maneja os arranjos em memória virtual apresentam a vantagem de estarem escritos em um FORTRAN altamente compatível, o que facilita sua tradução para qualquer computador, devendo somente obedecer-se àquelas instruções que têm a ver com o manejo do disco, as quais mudam de computador para computador.

Neste capítulo se descreve brevemente o funcionamento geral do sistema, os comandos em uso e diversos detalhes da implementação, que incluem as características do arquivo geral criado em disco. Descrevem-se também, com certo detalhe, as rotinas disponíveis na linguagem, úteis para a elaboração de novos comandos.

5.2 - CARACTERÍSTICAS GERAIS

A linguagem orientada é um módulo muito importante para o funcionamento do sistema, já que constitui a via de comunicação com o usuário, o qual, por este meio, organiza a base de dados (arquivo em disco que ficará à disposição dos módulos de análises) e a impressão de resultados, podendo ser seletivo quanto à impressão de todos eles ou somente de alguns.

Esta comunicação do usuário com o sistema se realiza

através de "comandos " definidos por certas palavras chave em es
panhol, tais como "TITULO", "COORDENADAS", "CONSTANTES", etc. Mais
adiante serão descritos com detalhes os comandos atualmente em
funcionamento.

Todos os dados podem ser proporcionados em formato li
vre. Além do mais, existe uma série de rótulos definidos den
tro de cada comando, que permite dar certos dados em uma ordem
arbitrária, como ocorre, por exemplo, com o comando "coordena -
das". Todo erro na sintaxe de um comando trará como consequên
cia a emissão de uma mensagem de erro, sem que por isso se inter
rompa a leitura dos dados. Os comando podem ser dados em uma
ordem arbitrária, a critério do usuário.

Os comandos atualmente em funcionamento são enumera -
dos como se segue:

TITULO
TIPO
COORDENADAS
CONECTIVIDADES
CONSTANTES
RESTRICCIONES NODALES
PROPIEDADES
CARGAS APLICADAS
SUPERELEMENTO
IMPRESION
ANALISIS
FIN

O comando "Título" serve para especificar o nome do
programa a ser resolvido. O comando "Tipo" apresenta as opções
"LAPLACE", "POISSON", "CAMPO", "CALOR", "FILTRAGEM", e "HEM-
HOLTZ" e determina o tipo de problema a ser analisado.

A definição da geometria da malha se realiza através
dos comandos "COORDENADAS" e "CONECTIVIDADES". As coordenadas de
um nó podem ser especificadas em qualquer ordem, se são precedi
das por rótulos "x" ou "y", de acordo com o caso. Ademais o pro

grama leva uma numeração interna dos nós e dos elementos, segundo a ordem pela qual são introduzidos. A equivalência entre esta numeração interna e dos nomes que proporciona o usuário, se realiza através de arranjos, onde estes últimos são armazenados.

O comando "CONSTANTES" se refere a características comuns a todos os problemas. Atualmente se incluem neste comando os rótulos correspondentes às constantes que interferem na equação de campo: "KX", "KY", "LAMBDA", "C" (ver equação nº 4.2) e que têm diferente significado físico, segundo o problema que vai ser resolvido. Se não se fornecem os rótulos, o programa o programa assinalará os valores e as constantes, de acordo com a ordem que foi mencionada.

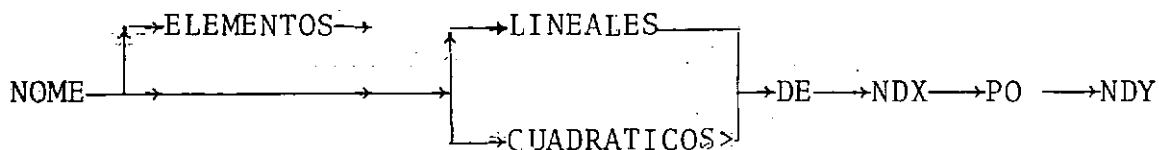
O comando "PROPIEDADES" permite assinalar diversas características para cada elemento em particular, o que facilita, por exemplo, considerar vários tipos de material em um mesmo problema. A ordem com que registra estas propriedades corresponde aos rótulos disponíveis: "KX", "KY", "Q1", "Q2", "Q3", "H1", "H2", "H3", "T1", "T2" e "T3", onde os dois primeiros representam, quantidades que aparecem na equação de campo e os restantes cobrem todos aqueles valores necessários para definir condições de fluxo e de convecção em alguns problemas de calor (ver equações 4.6 e 4.14)

O comando "RESTRICCIONES NODALES" é utilizado para especificar valores dados de uma função em estudo em nós escolhidos segundo o problema. Assim sendo, o comando "CARGAS APLICADAS" é empregado naqueles problemas que apresentam cargas aplicadas em seus nós. Os problemas de aplicação da equação de campo (até agora único módulo de aplicação implementado) não fazem uso deste comando.

Nos três últimos comandos que foram mencionados pode se fazer uso da facilidade de registrar um mesmo valor a uma lista de nós ou elementos, ao invés de repetir a instrução para cada um deles.

O comando "SUPERELEMENTO" permite a geração de ma-

lha em recintos quadriláteros de lados curvos e retos. Os elementos finitos gerados podem ser triângulos isoparamétricos de primeira e segunda ordem. O comando é esquematizado a seguir.



O nome identifica o superelemento que vai ser gerado. Os inteiros NDX e NDY especificam o número de divisores que vão ser efetuados sobre os lados do quadrilátero. Os vértices, ou pontos que definem o recinto, são enumerados em um cartão, em seguida, sendo descritos segundo suas coordenadas. Estes pontos podem ser em número de quatro ou oito, se o recinto é de lados retos ou de lados parabólicos.

Este esquema de geração se baseia no conceito de elemento isoparamétrico [19]. Consequentemente, considerando o caso particular de um quadrilátero parabólico (Fig.5.1) do qual se conhecem as coordenadas dos 8 nós que o definem, as coordenadas cartesianas de qualquer ponto em seu interior podem ser definidas em função das primeiras, da seguinte forma:

$$\begin{aligned}
 X &= \sum_{i=1}^8 N_i X_i \\
 Y &= \sum_{i=1}^8 N_i Y_i
 \end{aligned}
 \tag{5.1}$$

onde as N_i são funções de forma associadas a cada nó e definidas em termos de um sistema de coordenadas curvilíneo (ϵ, η) , cujos valores oscilam entre (-1) e (1) .

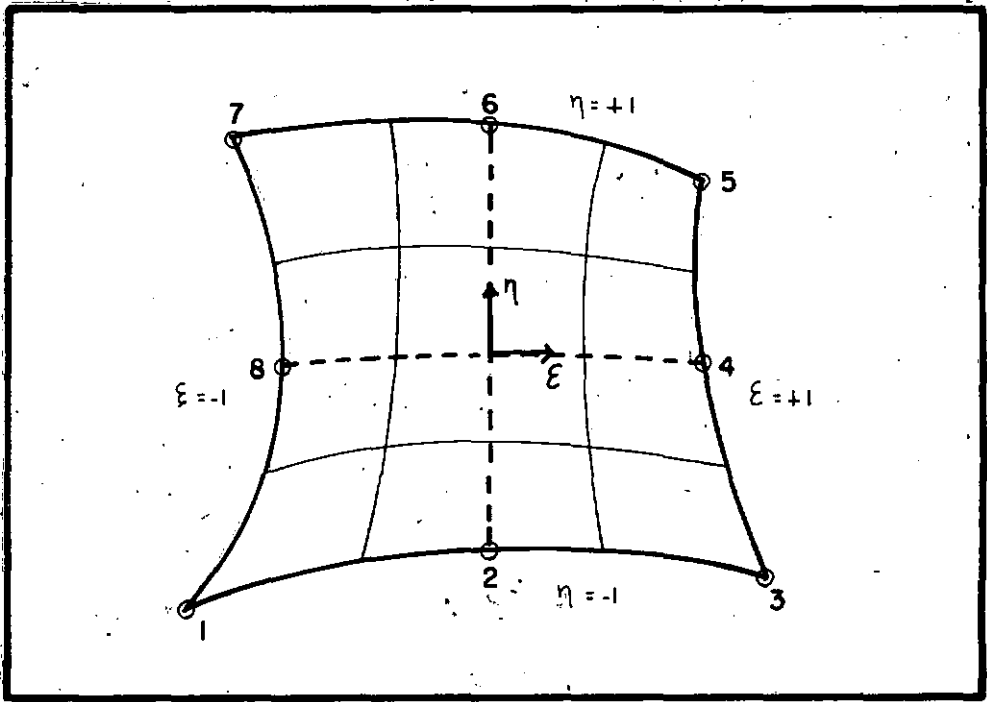


Figura 5.1
Quadrilátero Parabólico

As funções de forma são:

$$\begin{aligned}
 N_1 &= -\frac{1}{4} (1-\xi) (1-\eta) (\xi+\eta+1) \\
 N_2 &= \frac{1}{2} (1-\xi^2) (1-\eta) \\
 N_3 &= -\frac{1}{4} (1-\eta) (1+\xi) (\eta-\xi+1) \\
 N_4 &= \frac{1}{2} (1-\eta^2) (1+\xi) \\
 N_5 &= \frac{1}{4} (1+\xi) (1+\eta) (\xi+\eta-1) \\
 N_6 &= \frac{1}{2} (1-\xi^2) (1+\eta) \\
 N_7 &= \frac{1}{4} (1-\xi) (1+\eta) (\eta-\xi-1) \\
 N_8 &= \frac{1}{2} (1-\eta^2) (1-\xi)
 \end{aligned} \tag{5.2}$$

Segundo o número de divisões especificado por NDX e NDY, uma série de quadriláteros se criam dentro do recinto, os quais são divididos em triângulos segundo sua diagonal mais cur

ta: isto proporciona triângulos mais bem conformados.

Os nomes dos nós que constituem um super elemento são registrados sequencialmente a partir de um número base que contempla os nós anteriormente gerados. Quando se declaram dois elementos adjacentes, a numeração dos nós comuns que se conserva é correspondente ao lado do primeiro super elemento declarado. Isto pode provocar um considerável incremento na largura de banda da matriz de rigidez, se não se utiliza racionalmente o comando. Recomenda-se então, tomar precauções quanto à ordem em que se geram os super elementos, e também, quanto à forma como se apresentam os laos (ordem em que se proporcionam os nós que definem o super elemento). Um exemplo disto é mostrado na Figura 5.2)

Na realidade se projeta para um futuro imediato a implementação de algoritmos de renumeração que minimizarão a largura de banda, fazendo com que o uso deste comando seja mais eficiente; e também a condensação dos super elementos em seus nós de contorno para permitir uma análise por sub-estruturas que amplie a capacidade do sistema.

É conveniente assinalar, também, as restrições que existem na aplicação deste esquema de geração de malhas, baseado em coordenadas isoparamétricas no que se refere à colocação do nó central.

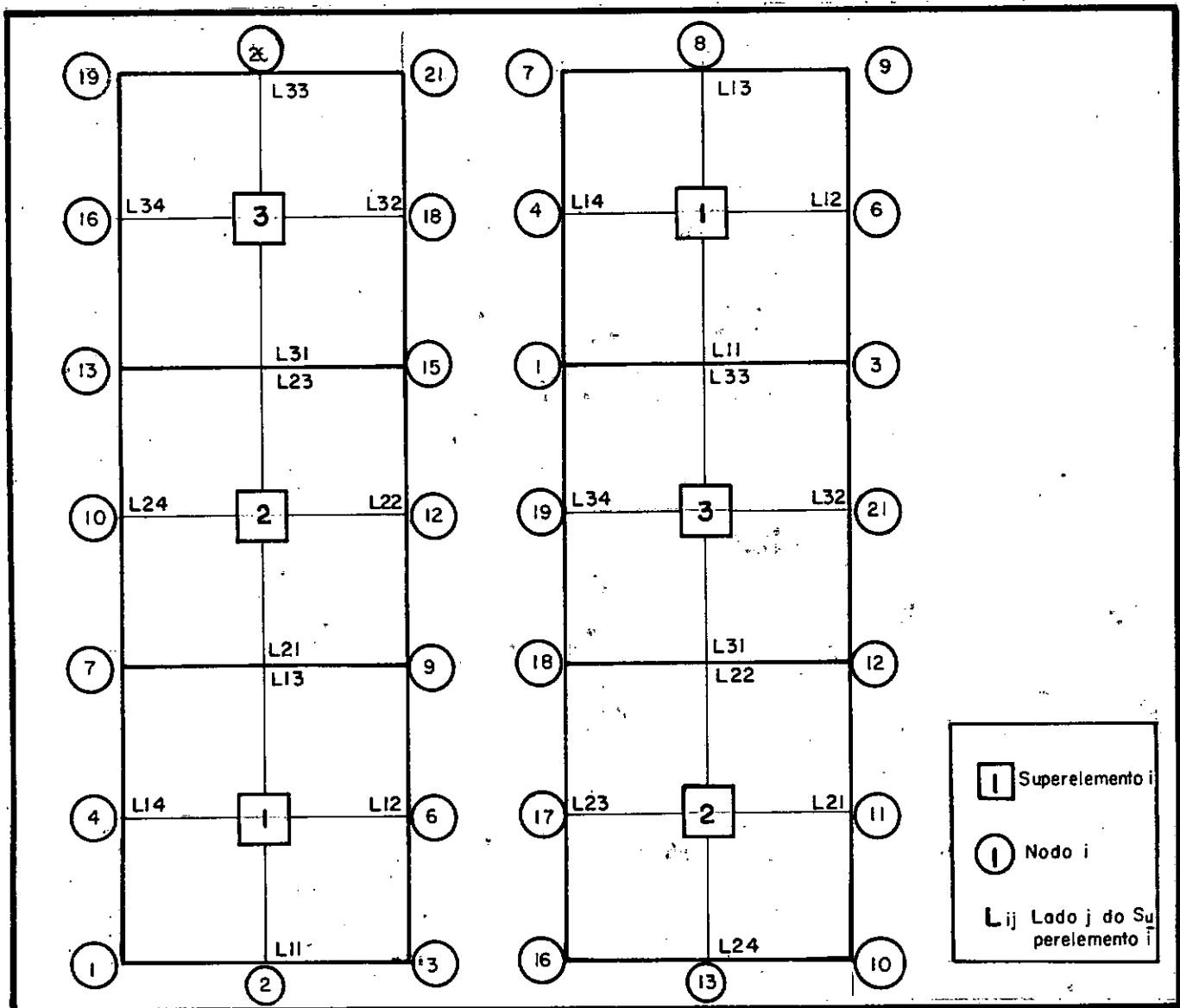


Figura 5.2

Utilização do Comando Superelemento

Para a malha da esquerda a diferença nodal máxima é de 4; para a da direita é de 17.

Na realidade, é sabido que selecionando o posicionamento dos pontos centrais sobre os lados, pode se obter malhas mais ou menos densas em diferentes regiões do recinto (Figura 5.3). No entanto, o posicionamento destes pontos não é arbitrário, e deve se limitar a determinada região sobre o lado; já que do contrário podem ocorrer malformações na malha gerada: pontos fora do recinto, linhas que se cruzam, etc. Recomenda-se, para assegurar que isto não suceda, que o ponto central se encontre separado do vértice por uma distância que não seja menor que um quarto do lado, e não seja maior que três quartos dele. Na realidade, (ver Ref. 20) os valores destas distâncias dependem do número de pontos que vão ficar sobre o lado:

$$\begin{aligned} \frac{B}{L} &= \frac{(N-3)}{4(N-2)} \\ \frac{B}{L} &= \frac{(3N-5)}{4(N-2)} \end{aligned} \quad (5.3)$$

onde L é o comprimento do lado, B é a distância em relação ao vértice, e N o número de nós do lado.

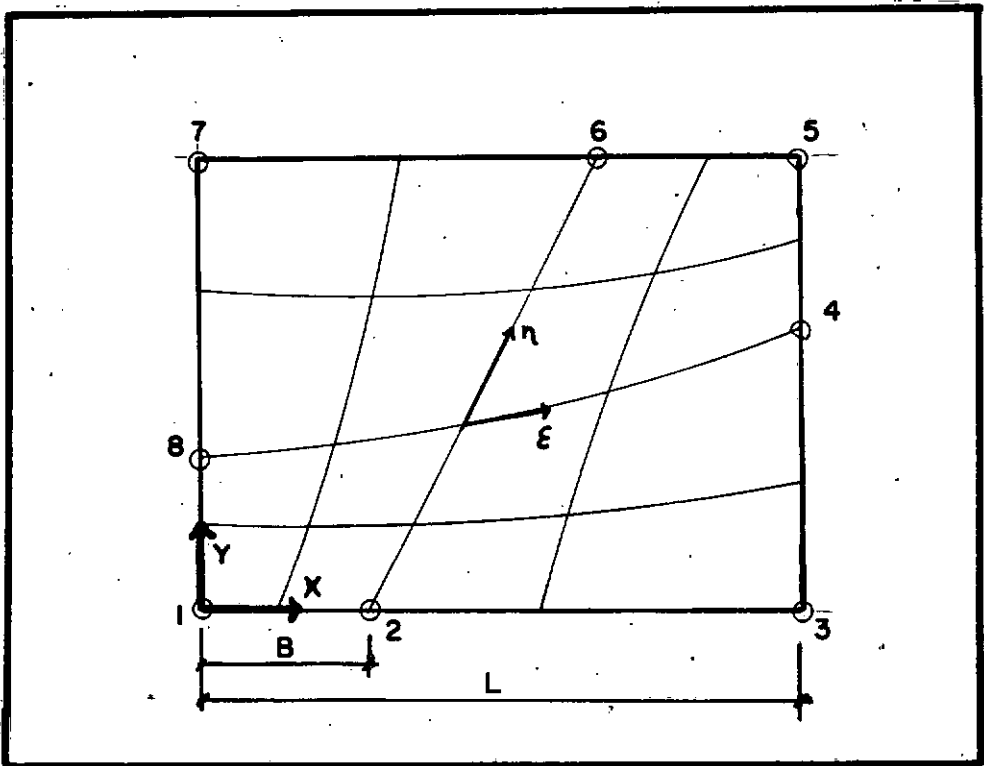
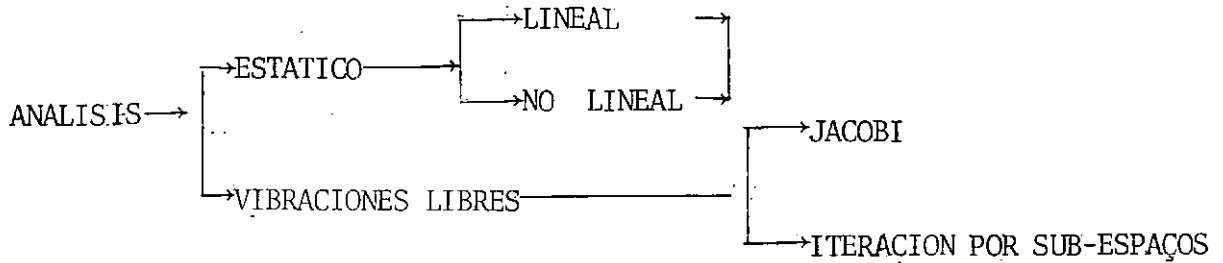


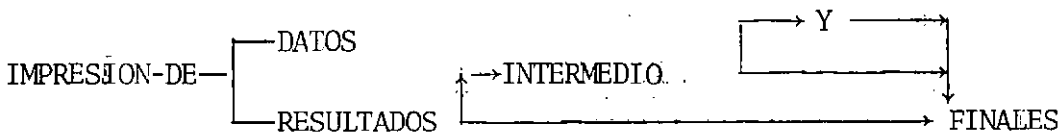
Figura 5.3
Efeito do Refinamento na Malha Gerada

O comando "ANALISE" define o tipo de análise que vai ser realizada por outros módulos do sistema. A estrutura deste comando pode ser representada da seguinte forma:



A estrutura atual deste comando abrange todos os tipos de análises que o sistema realiza. Posteriormente, a medida que se vão implementando, se ampliará o comando para cobrir novos tipos de análises.

O comando "IMPRESSÃO" se esquematiza como se segue:



Se não se indicam dados ou resultados específicos, são todos eles impressos; do contrário, o usuário pode selecionar alguns deles unicamente, os quais coloca no cartão ou cartões seguintes. O comando "impressão de resultados" ativa uma série de indicadores que serão utilizados após a análise para selecionar aqueles que serão impressos. Estes podem se referir ao valor da função nos nós e/ou ao valor das derivadas da função sobre os elementos. Indicam-se resultados intermediários quando se deseja conhecer os valores através da cada iteração.

O comando "FIM" deve ser o último especificado na entrada de dados. Este comando indica a reorganização do arquivo e a eliminação daqueles arranjos que não necessários. O arquivo permanece em Disco com o nome de "DADOS" e pode ser aberto por qualquer outro programa de aplicação e/ou análise.

5.3 - ALGUNS DETALHES SOBRE A IMPLEMENTAÇÃO

A linguagem orientada implementada permite evitar ao usuário toda preocupação quanto ao uso do disco como memória suporte. já que se apoia na utilização de um software de memória virtual especialmente concebido para minicomputadores [13] e [14]. O uso deste sistema permite a declaração de grandes arranjos, sem as limitações impostas pela pouca quantidade de memória direta disponível.

Estes arranjos, em princípio são destinados a coordenadas, conetividades, propriedades, nomes dos elementos e dos membros, cargas e condições de contorno. A multiplicidade dos arranjos permite maior flexibilidade no uso dos comandos. Todavia, quando se introduz o comando "FIN", ocorre uma reorganização geral. A maioria dos dados é disposta em um único arranjo denominado área, de duas dimensões, o qual apresenta um aspecto como se vê na Fig. 5.4, que mostra uma fileira do arranjo AREA. Este arranjo tem tantas fileiras quanto números de elementos haja.

ELEMENTO	NODO I			NODO J			NODO K			FLUJO			CONVECCION		
100	I	XI	YI	J	XJ	YJ	K	XK	YK	KX	KY	Q1	Q2	Q3	H1 H2 H3

Figura 5.4
Fila do arranjo AREA

Além destes se conservam os nomes dos nós, (já que no arranjo AREA só figuram os números correspondentes à sua numeração interna) e informação geral do problema como: número de nós, número de elementos constantes, título, tipo de análise a realizar, condições de contorno, etc. Há outro tipo de informação que é inerente ao sistema de memória virtual (tamanho da gravação, número de páginas, tamanho da página, etc.) que também é armazenada para poder entrar posteriormente no arquivo. No entanto, não se menciona isto, porque o usuário nunca tem que manejar este tipo de informação. Os demais arranjos são "apagados"

do disco, por não serem necessários para os módulos de análise.

A organização especial que se mostra para o arranjo AREA, foi selecionada visando a eficiência do sistema. Assim sendo, ainda que desta forma se armazene várias vezes a mesma informação, (como é o caso das coordenadas dos nós) se minimizam os acessos ao disco na hora de montagem da matriz de rigidez, já que, ao colocar uma fila do arranjo em memória direta, se têm todos os dados necessários para a construção da matriz do elemento.

O funcionamento geral da linguagem orientada se baseia na utilização de um "dicionário", que traduz os caracteres alfanuméricos fornecidos pelos comandos, para caracteres numéricos. Assim, por exemplo, o "A" corresponde ao nº 16, o "J" ao número 15, etc. A tabela 5.1 mostra os caracteres disponíveis no dicionário e seu código numérico.

Quando um cartão é lido, passa a ser interpretado, de forma que todas suas colunas, cheias de caracteres alfanuméricos se traduzam em um arranjo equivalente de caracteres numéricos. Este trabalho é realizado por uma subrotina chamada LEER.

Esta subrotina também se encarrega de organizar a informação, de forma que, por exemplo, estabeleça o número de palavras ("NES") que exista em um comando; cria um arranjo ("IPT"), que, contendo duas posições por cada palavra, na segunda indica si se trata de uma palavra numérica ou alfanumérica, e na primeira, a posição da primeira coluna da palavra, dentro do comando. Esta subrotina discrimina também o número de comandos, separados por ";" ou por "%", que se encontram em cada cartão e leva a informação da coluna na qual começam e em qual terminam. Desta forma, ainda que uma chamada LER signifique a interpretação de um novo comando, não significa necessariamente a leitura de um novo cartão. A Figura 5.5 mostra um diagrama de fluxo esquemático da subrotina LER. Esta subrotina utiliza os serviços de uma função, que segundo determinado código, (1 ou

CÓDIGO NUMÉRICO	CARACTER	CÓDIGO NUMÉRICO	CARACTER	CÓDIGO NUMÉRICO	CARACTER
1	b	17	B	33	
2	0	18	C	34	S
3	1	19	D	35	T
4	2	20	E	36	U
5	3	21	F	37	V
6	4	22	G	38	W
7	5	23	H	39	X
8	6	24	I	40	Y
9	7	25	J	41	Z
10	8	26	K	42	C
11	9	27	L	43	%
12	-	28	M	44	/
13	+	29	N	45	(
14	.	30	O	46)
15	;	31	P		
16	A	32	Q		

Tabela 5.1
Caracteres do Dicionário

2) busca o primeiro caracter igual, ou primeiro diferente do determinado por um dos parâmetros. Isto é realizado a partir de certa posição fornecida segundo as necessidades, dentro dos limites do comando. IBUS, o nome da função, devolve a posição do caracter procurado.

Uma vez organizada a informação dada por um cartão, procura-se indentificar qual o comando que contém. Isto se realiza por pesquisas, comparando os primeiros caracteres codificados a partir do cartão, com as três ou quatro primeiras letras de cada uma das palavras que definem um comando todas as possibilidades que possui a linguagem. Se nenhum comando é conseguido, emite-se uma mensagem de erro e passa-se para o próximo comando.

Uma vez identificado o comando, procede-se o armazenamento dos seus dados, dispondo-se para isso de uma série de subrotinas que realizam diversas funções. Estas subrotinas são chamadas: DANO, LISTA, LISRO, que por sua vez se servem das rotinas NUMER e FINDI.

A forma de utilização dessas subrotinas, depende do comando específico que se está desenvolvendo. Neste sentido, existem dois tipos diferentes de comandos, os que têm rótulo e os que não têm.

Os comandos com rótulo são, dentre os que foram mencionados anteriormente, aqueles nos quais a ordem específica dos seus dados pode ser alterada, se estes dados forem identificados com seu respectivo rótulo. Para este tipo de comando, deve se providenciar um arranjo onde se identifique, por uma ou duas letras, todos aqueles rótulos que podem aparecer na sua utilização. Este arranjo é então fornecido à rotina DANO, que se encarrega de armazenar dados identificados por rótulos. Exemplificando, o arranjo de rótulos correspondente ao comando "COORDENADAS" seria da seguinte forma:

$$IR = \begin{bmatrix} X \\ Y \end{bmatrix} \quad (5.4)$$

ou o que da no mesmo : (ver Tabela 5.1)

$$IR = \begin{bmatrix} 39 \\ 40 \end{bmatrix} \quad (5.5)$$

A subrotina "DARO", na realidade, utiliza os serviços de "LISTA" e "LISTRO", e, segundo os dados que obtém, é capaz de identificar uma das seguintes situações, devolvendo um indicador adequado:

- os dados no cartão foram colocados em um arranjo auxiliar segundo uma ordem conhecida (estabelecida pela ordem em que se colocam os rótulos no arranjo correpondente) e os nós ou elementos aos quais corrépondem estes dados, se encontram definidos em outro vector auxiliar;
- conseguiu-se no cartão um rótulo não considerado no arranjo de rótulos;
- imprimiu mensagem de erro, por alguma situação incorreta;
- a primeira palavra no cartão não é um número.

Na definição destas situações intervem a subrotina LISTA, atuando em uma das duas formas:

- preenchendo um arranjo com uma lista sequencial de nomes (código 2);
- preenchendo um só lugar no arranjo (código 1).

Este último tem sentido quando se trabalha com comandos como "COORDENADAS", onde não é possível registrar uns mesmos valores de coordenadas X e Y a uma lista de nomes. Neste caso a subrotina averigua se há ou não mais caracteres do cartão para posteriormente passar os dados a "LISRO". Além do mais indica o número da palavra na qual ficou pendente. A Figura 5.6 mostra o dia-

grama de fluxo da subrotina "LISTA".

A subrotina "LISRO" retoma o comando na palavra indicada por "LISTA", e é encarregada de anotar os dados indicados no cartão (seja com rótulos ou sem êles) em um arranjo auxiliar. Os dados são organizados especificamente, de acordo com o aparecimento dos rótulos no arranjo de rótulos. (Fig. 5.7).

Existem alguns comandos com rótulo que nunca fazem uso de uma lista de nomes, porque podem utilizar diretamente a subrotina "LISRO", sem chamar previamente a "DARO" e a "LISTA". Tal é o caso do comando "CONSTANTES", por exemplo.

As subrotinas anteriores utilizam a subrotina "NUMER" para identificar uma quantidade determinada, quer seja um número inteiro real, quer seja em forma exponencial. Assim sendo, devolve a posição da coluna vazia posterior ao número e um código de erro. O funcionamento desta rotina é explicado esquematicamente na Fig. 5.8.

Uma vez que, nos cartões com rótulo, forem identificados os dados correspondentes, e com a ajuda das subrotinas anteriormente descritas, forem armazenados convenientemente em arranjos auxiliares tanto os mencionados dados, como a lista de nomes ou de elementos, às quais estes dados correspondem, coloca-se esta lista de dados em seu respectivo arranjo, o qual se encontra armazenado no disco. Todavia, para armazenar os dados de acordo com a numeração interna que utiliza o programa, é necessário efetuar a comparação entre esta numeração e a lista de nomes fornecida pelo usuário. Esta tarefa é realizada pela subrotina "FINDI".

Em resumo, a elaboração de um comando qualquer, envolvendo o registro de dados, com ou sem rótulos, a um nome ou a uma lista de nomes, é esquematizada nas seguintes etapas:

- 1) Identificação do comando.
- 2) Elaboração do arranjo de rótulos próprios do comando.

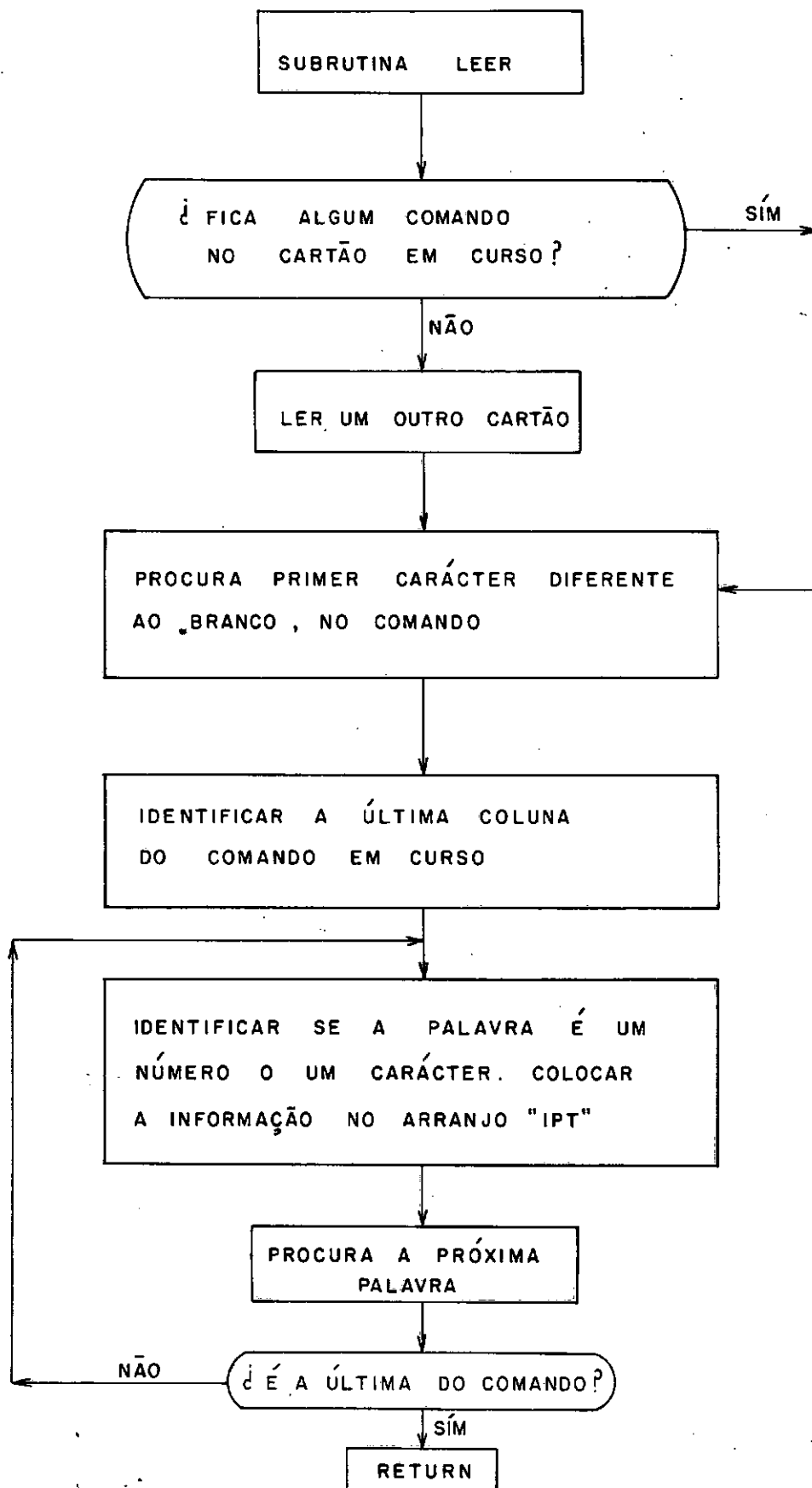


Figura 5.5
Subrotina LER

SUBROTINA LISTA

SE O CÓDIGO É UM , ENCHE SÓ UMA POSIÇÃO NO ARRANJO "LIS". SE É 2 ENCHE MAIS DO QUE UM

INÍCIO

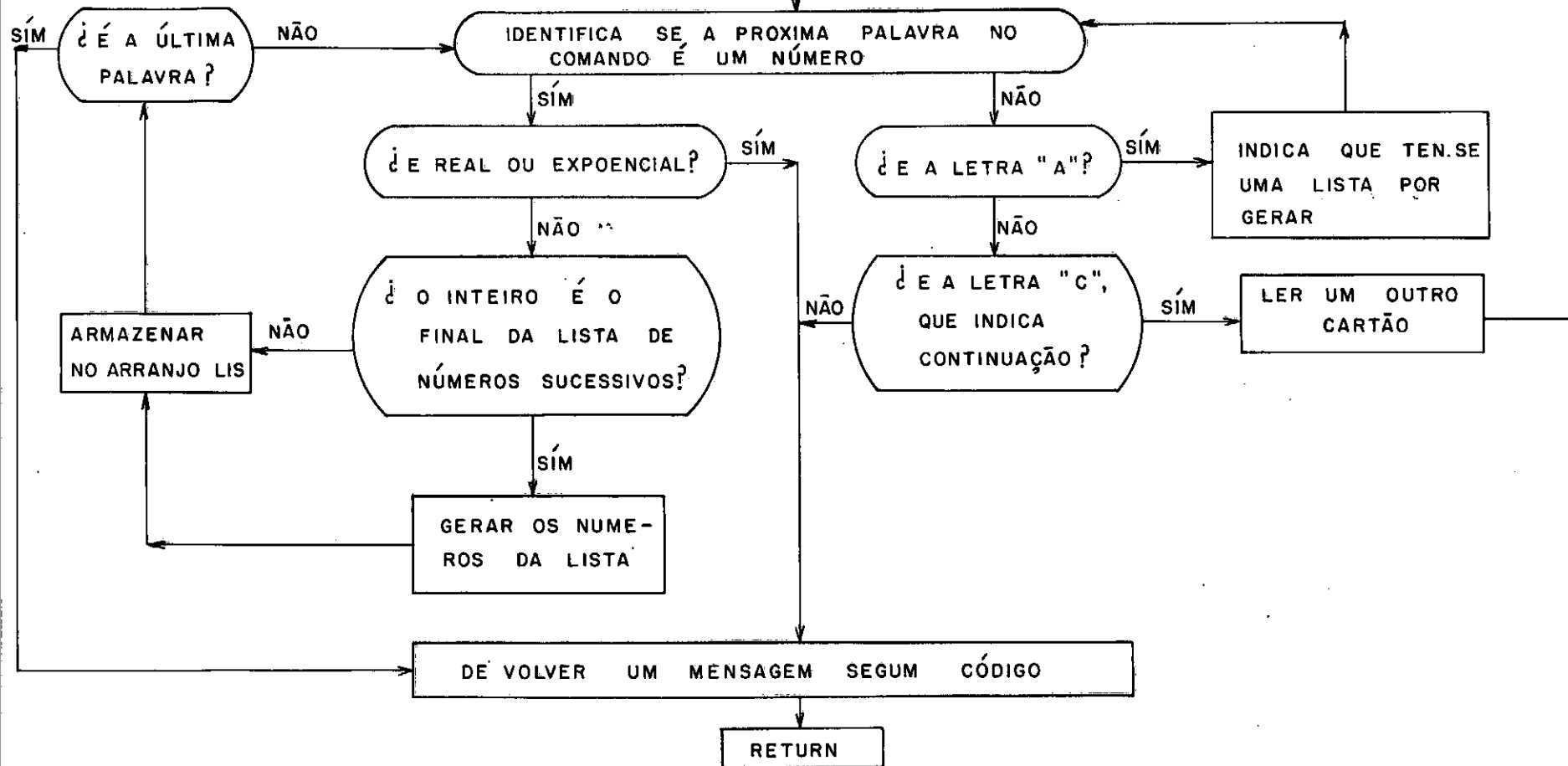


Figura 5.6
Subrotina LISTA

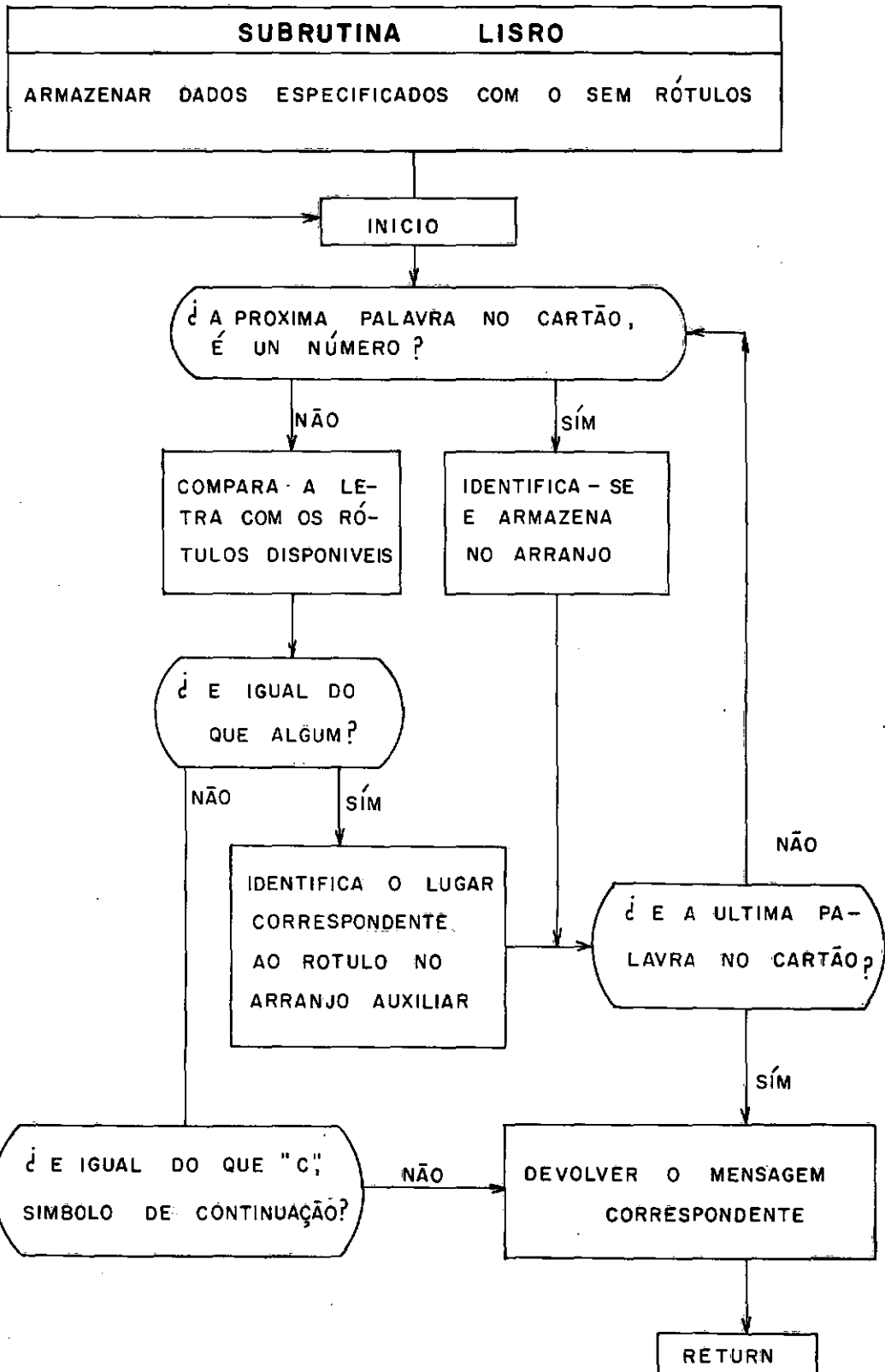


Figura 5.7
Subrotina LISRO

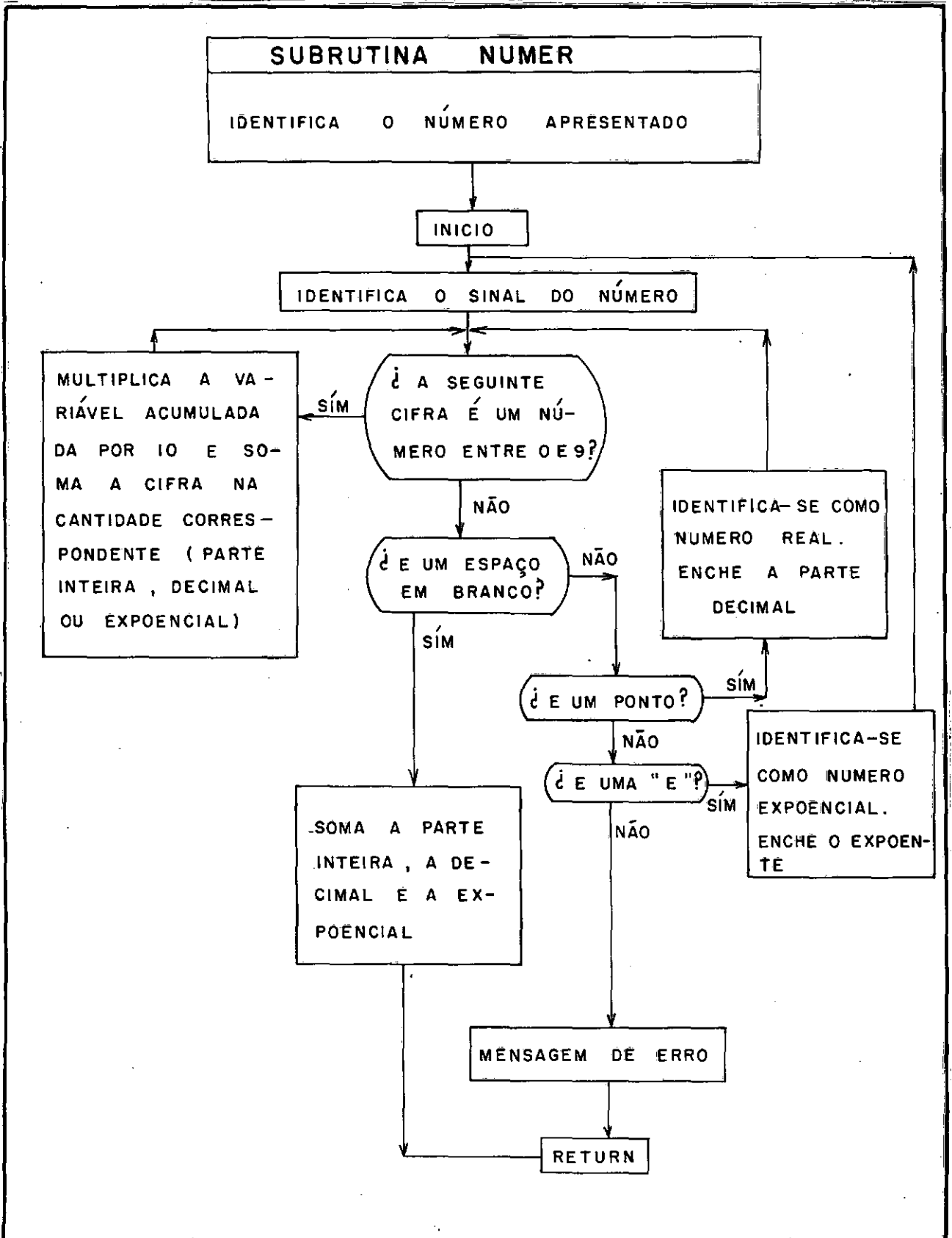


Figura 5.8

Subrotina NUMER

- 3) Identificação e construção de uma lista de nomes.
- 4) Identificação dos dados e registro em um pequeno arranjo auxiliar, em memória direta.
- 5) Registro dos dados no arranjo maior em disco, de acordo com a numeração interna que corresponde ao nomes dados. (Ver Fig. 5.9)

Os comandos sem rótulo, como os de "CONETIVIDADES", "TIPO", "IMPRESSAO", etc., utilizam algumas das subrotinas mencionadas de forma muito particular, segundo as necessidades de cada comando e aproveitando os recursos disponíveis.

Assim, por exemplo, o comando "CONETIVIDADES" utiliza a subrotina "LISTA" para armazenar todos os dados de um elemento em um arranjo auxiliar. Posteriormente, usa a subrotina "FINDI", para colocar em cada elemento suas conexividades, visando a equivalência dos nomes e dos nós que a integram com a numeração interna destes nós. Entende-se que todo cartão é interpretado pela subrotina "LER".

Concluindo, a adição de um novo comando aos já disponíveis pela linguagem orientada, é facilitada pela existência de um número adequado de subrotinas de serviço, que se ocupam de apresentar adequadamente a informação de um cartão. É imprescindível que se conheça a capacidade destas subrotinas e os parâmetros necessários para o seu funcionamento (Tabela 5.2), com a finalidade de permitir a implementação eficiente de novos comandos que facilitem a entrada de dados. Atualmente não estão implementados no sistema, comandos que se ocupem de organizar os resultados, proporcionando facilidades de saída, mas projeta-se utilizar a filosofia da linguagem atual para acrescentar saídas gráficas (projetos em plotter e/ou PRINTER), etc.

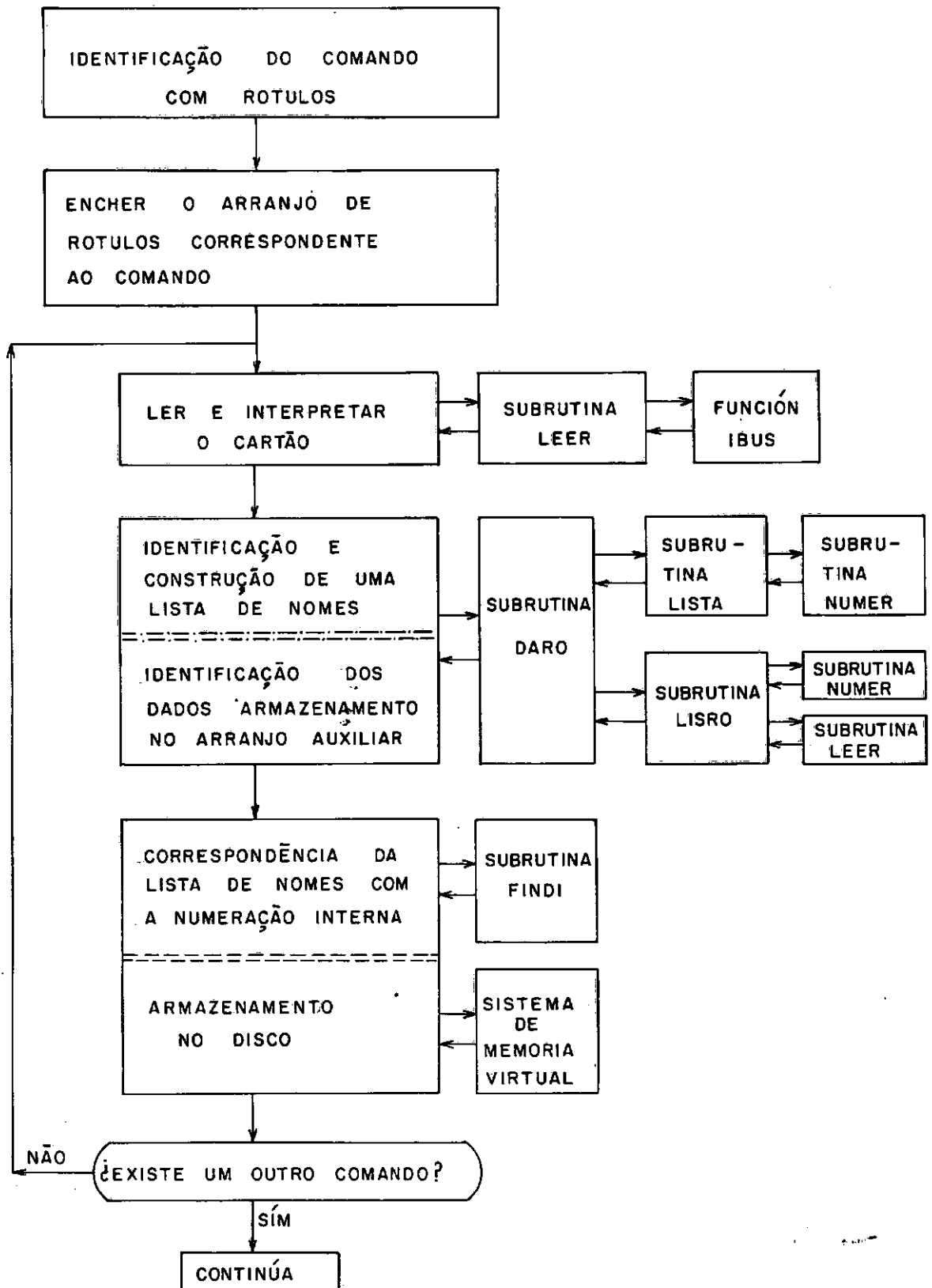


Fig 5.9
Comandos com Rótulos

SUBROTINA	UTILIZAÇÃO	PARÂMETROS	SIGNIFICADO DOS PARÂMETROS
DARO	Armazena dados com ou sem rótulos, referentes a um nome ou a uma lista de nomes.	IR DA LIS N NL KØD	Fornece os rótulos que o comando requer Arranjo Auxiliar: devolve dados Arranjo Auxiliar: devolve nomes Número total de rótulos Devolve número de nomes em LIS 1: Um só nome 2: Lista de nomes
LIBRO	Interpreta os dados, com ou sem rótulos, que encontra no comando.	IR DA N	VER SUBROTINA DARO
LISTA	Armazena uma lista de nomes segundo um código.	N KØD	VER SUBROTINA DARO
FUNCTION IBUS	Procura caracteres. Devolve a posição do nome.	IK IDIG II	1: Busca caracter igual a: 2: Busca caracter diferente de: Código do Caracter Busca a partir da coluna II
NUMER	Interpreta o número.	IQ IZO R NENT	Devolve a colocação da coluna posterior Devolve o código: 1= número inteiro 2= número decimal 3= forma exponencial 4= erro Devolve o número em forma real Devolve a parte inteira
LER	Interpreta o conteúdo de um cartão	NÃO TEM	
FINDI	Busca um número em um arranjo. se não existe o coloca no final.	NUE NAT NUI KINO	Número a ser procurado Número total de elementos em um arranjo Devolve a posição de NUE Arranjo onde se procura

TABELA 5.2
Subrotinas Úteis,
de Propósito Geral

CAPÍTULO VI

PROCEDIMENTOS DE ANÁLISES

6.1 - INTRODUÇÃO

Os procedimentos de análises são ferramentas matemáticas aplicáveis a numerosos problemas de engenharia. Estas ferramentas são gerais, já que não dependem, no seu esquema global, do problema que está sendo resolvido. Isto permite que os módulos de análise mostrados na Fig. 3.1, que constituem o sistema, e que neste trabalho são aplicados a problemas referentes à equação de campo (Capítulo IV), possam ser utilizados de forma geral, para aplicações na área de estruturas.

Neste capítulo são descritos os métodos implementados para análises linear, não linear e problemas de autovalores e autovetores.

6.2 - ANÁLISE LINEAR

Na realidade, esta constitui a solução de um sistema de equações não dependente do tempo, do tipo:

$$\underline{K} \underline{\phi} = \underline{P} \quad (6.1)$$

onde a matriz \underline{K} de "rigidez", é independente, o mesmo acontecendo com as "cargas", \underline{P} , do vetor das incógnitas, $\underline{\phi}$.

Isto se realiza com base em um sistema de triangularização de Gauss, onde a matriz \underline{K} se descompõe em:

$$\underline{K} = \underline{L} \underline{D} \underline{L}^T \quad (6.2)$$

sendo \underline{D} uma matriz diagonal e \underline{L} uma matriz triangular inferior [27]. A matriz \underline{K} se encontra armazenada aproveitando as suas características de banda e simetria, de forma convencional.

A implementação efetuada permite a resolução de vários casos de carga, simultaneamente, e o tempo de acesso ao disco se minimiza, porque os vetores de carga são armazenados ao lado da matriz \underline{K} .

Como o maior tempo empregado em um procedimento de análise é a quota correspondente à solução do sistema de equações, e neste sentido os procedimentos descrito neste Capítulo não são mais que aplicações reiteradas da análise linear, isto é, resoluções repetidas de diferentes sistemas de equações construídos segundo características próprias de cada método, adotou-se uma estratégia especial no manejo da informação contida no disco. Esta estratégia já foi explicada no capítulo III e consiste em conservar em memória direta, pelo tempo que se vai utilizar para reduzir as demais filas, a fila que já foi de finida como fila "pivote", tanto no processo de triangularização como na redução dos vetores de carga. A diferença nos tempos utilizando-se o esquema de resgate de informação segundo a fila pivote, em relação à estratégia primitiva, que não a utilizava, é considerável. [11]

Outra forma que foi empregada para diminuir o esforço computacional no caso de resoluções de sistemas sucessivos, utilizando uma matriz \underline{K} , foi o aproveitamento do esquema particular de triangularização utilizado (Eq. 6.2), o qual implica na resolução de dois sistemas auxiliares, um por substituição "para frente" com a matriz triangular inferior e outro por retro-substituição, com sua transposta

$$\underline{L} \underline{Y} = \underline{P} \quad (6.3)$$

$$\underline{D} \underline{L}^T \underline{\phi} = \underline{Y} \quad (6.4)$$

Este esquema é útil, já que economiza o esforço computacional ao evitar o processo de triangularização repetido da matriz \underline{K} , quando se deseja resolver várias vezes um sistema de equações com base na mesma matriz. Tal é o caso de alguns métodos de análise não linear que utilizam uma mesma matriz \underline{K} para todas as iterações, e o caso de análises de i

terações por sub-espacos.

6.3 - ANÁLISE NÃO LINEAR

Existem numerosos problemas em engenharia civil que correspondem a comportamentos não lineares. Tais são os casos de: aplicações em estruturas, problemas de não linearidade geométrica, não linearidade física e não linearidade nas relações deslocamento-deformação [18].

Para problemas de campo, a não linearidade pode se dar na dependência que possam ter as constantes K ou C , ou inclusive nas condições de contorno, do potencial desconhecido, ϕ , (ver seção 4.2). Dentro deste tipo de problemas, encontram-se, por exemplo, os de ignição espontânea [18], os de transmissão de calor em condições não lineares, problemas de percolação em meios porosos, onde não se cumpre a lei de Darcy [23], problemas de magnetismo onde a permeabilidade magnética depende das densidades de fluxo [22], etc.

Devido à grande quantidade de casos nos quais é aplicável uma análise não linear, é conveniente contar com algumas técnicas, das várias existentes, que a tornem possível. Estas técnicas podem ser do tipo incremental, ou passo a passo, do tipo iterativo, como o método de Newton-Raphson e derivados, e técnicas mixtas, que combinam as anteriores.

Nesta seção se descrevem brevemente diversos tipos de técnicas, fazendo-se um resumo de suas características e uma rápida discussão de suas vantagens e desvantagens.

Em geral, estas técnicas resolvem problemas do tipo:

$$K(\phi) * \phi = P \quad (6.5)$$

onde a matriz de coeficientes K , apresenta dependência com relação às incógnitas ϕ , em contraposição com o caso dos problemas lineares dados por:

$$K * \phi = P \quad (6.6)$$

Cabe mencionar neste ponto, que uma boa aplicação das técnicas para resolução de problemas não lineares não pode ser feita às cegas, mas deve se apoiar no conhecimento do comportamento físico do problema, já que estas técnicas podem conduzir, em certas ocasiões, a soluções múltiplas, dependendo do ponto de início, ou podem não conduzir a nenhuma solução em absoluto [18].

6.3.1 - Técnicas Incrementais para Análise Não Linear

Nestes tipos de procedimentos, o estado de "cargas" total desejado, P , é alcançado por sucessivos incrementos de carga, " ΔP ". Dentro de cada um destes incrementos é necessário resolver um sistema de equações lineares, para um valor determinado da matriz K , que depende dos resultados obtidos no intervalo anterior. Com isto se consegue o correspondente incremento na incógnita, $\Delta \phi$.

Para uma iteração " i ", o estado do problema virá dado por:

$$\{P\} = \{P_0\} + \sum_{j=1}^i \{\Delta P_j\} \quad (6.7)$$

$$\{\phi\} = \{\phi_0\} + \sum_{j=1}^i \{\Delta \phi_j\} \quad (6.8)$$

onde, P_0 e ϕ_0 são os valores iniciais, conhecidos, a partir dos quais se inicia o processo.

Para o intervalo " i " se utiliza o sistema de equações lineares:

$$[K_i - 1] \{\Delta \phi_i\} = \{\Delta P_i\} \quad (6.9)$$

donde a matriz $K_i - 1$ se constroi com base nos valores de iteração $(i-1)$:

$$[K_i - 1] = [K_i - 1 (\phi_{i-1})] \quad (6.10)$$

a matriz que geralmente se utiliza dentro do intervalo é a matriz tangente, expressada por:

$$[K_T] = \frac{d[P]}{d[\phi]} \quad (6.11)$$

A Figura 6.1 mostra graficamente o método explicado anteriormente

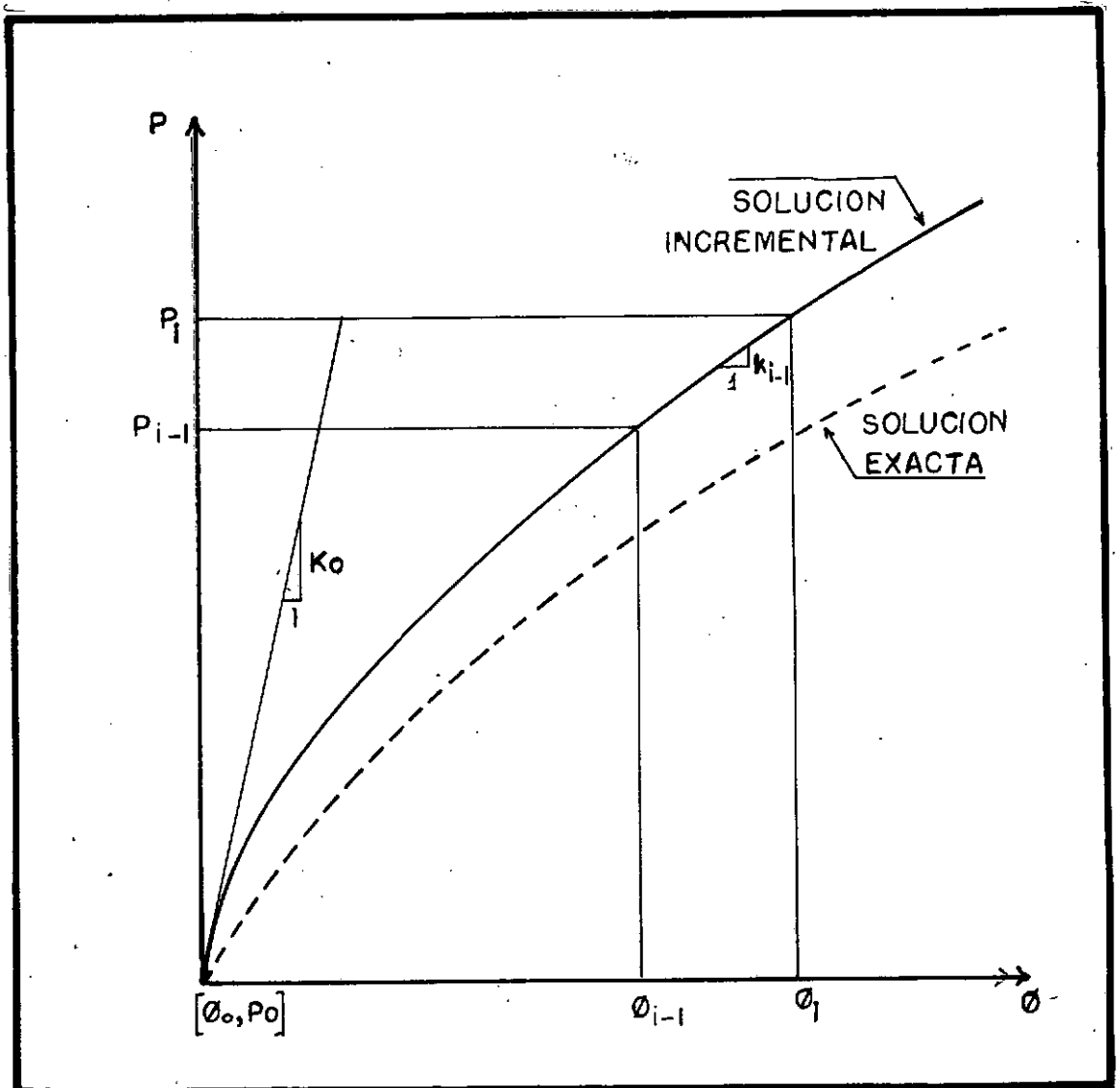


Figura 6.1

Procedimento Incremental

O método descrito corresponde a um método numérico utilizado para a integração de sistemas lineares e não-lineares: o método de EULER [23]. Pode-se garantir a convergência deste método para intervalos escolhidos P , o suficientemente pequenos. Pode-se melhorar o esquema de integração com a formulação de RUNGE-KUTTA para o ponto central do intervalo com um esforço computacional maior.

Para o método de RUNGE-KUTTA se calcula uma primeira aproximação do valor da função no ponto médio do intervalo, e, baseado neste, um valor da matriz K_T correspondente. As fórmulas para este caso são:

$$\left[K_i - 1 \right] \{ \Delta \phi_i - 1/2 \} = \frac{\{ \Delta P \}}{2} \quad (6.12)$$

$$\{ \phi_i - 1/2 \} = \{ \phi_i - 1 \} + \{ \Delta \phi_i - 1/2 \} \quad (6.13)$$

$$\left[K_{i-1/2} \right] = \left[K(\phi_i - 1/2) \right] \quad (6.14)$$

com o que se resolve o sistema

$$\left[K_{i-1/2} \right] \left[\Delta \phi_i \right] = \left[\Delta P_i \right] \quad (6.15)$$

Acumula-se o resultado:

$$\{ \phi_i \} = \{ \phi_i - 1 \} + \{ \Delta \phi_i \} \quad (6.16)$$

e com base nestes resultados se calcula uma nova matriz K e se inicia o processo novamente.

Uma das vantagens destes métodos é que proporcionam uma história completa do problema para diferentes estados de carga. Geralmente o custo destes métodos é maior que o dos métodos iterativos.

6.3.2 - Procedimentos Iterativos

Consistem em uma sequência de cálculos que se repetem para um único estado de cargas P, até alcançar a precisão requerida para o equilíbrio do sistema. Descreve-se a seguir, vários destes procedimentos:

a) Iteração Direta

Por este processo é também necessário resolver um sistema de equações lineares para cada iteração. No entanto, a matriz utilizada é a secante (Fig. 6.2), razão pela qual se adapta a problemas onde a curva de comportamento do material é dada por pontos, isto é, não se conhece a função explicitamente.

Para uma iteração "i" o procedimento a seguir seria:

$$[K_{i-1}] = [K(\phi_{i-1})] \quad (6.17)$$

e com este resolver o sistema de equações:

$$[K_{i-1}] \{\phi_i\} = \{P\} \quad (6.18)$$

Feito isto, calcula-se o "erro" na convergência:

$$e_i = \{\phi_i\} - \{\phi_{i-1}\} \quad (6.19)$$

que se compara com uma tolerância dada.

Este erro pode ser calculado de diversas formas

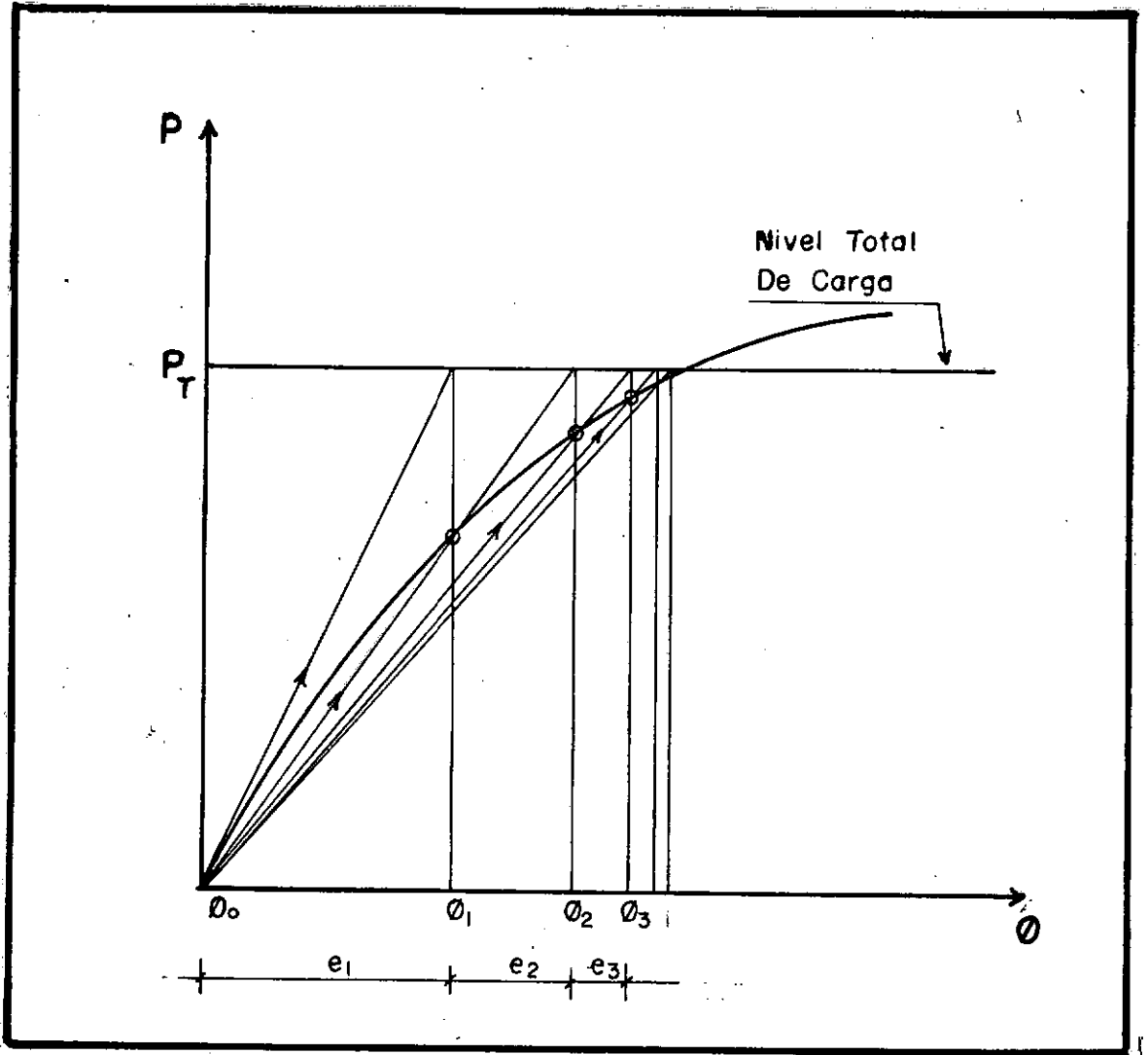


Figura 6.2
Iteração Direta

Este método é útil para problemas da equação de campo; todavia, existem ocasiões em que pode não convergir (Fig. 6.3)

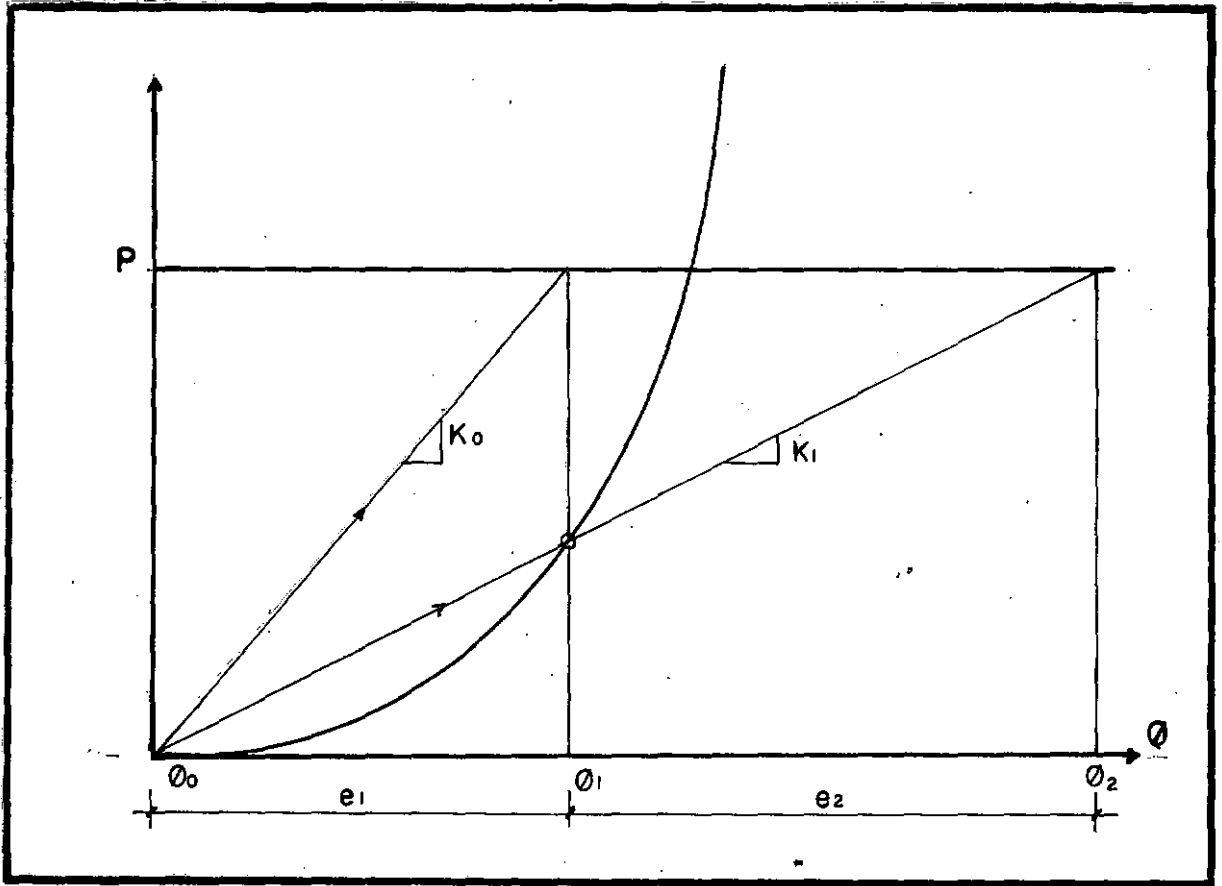


Figura 6.3
Divergência no Método de Iteração Direta

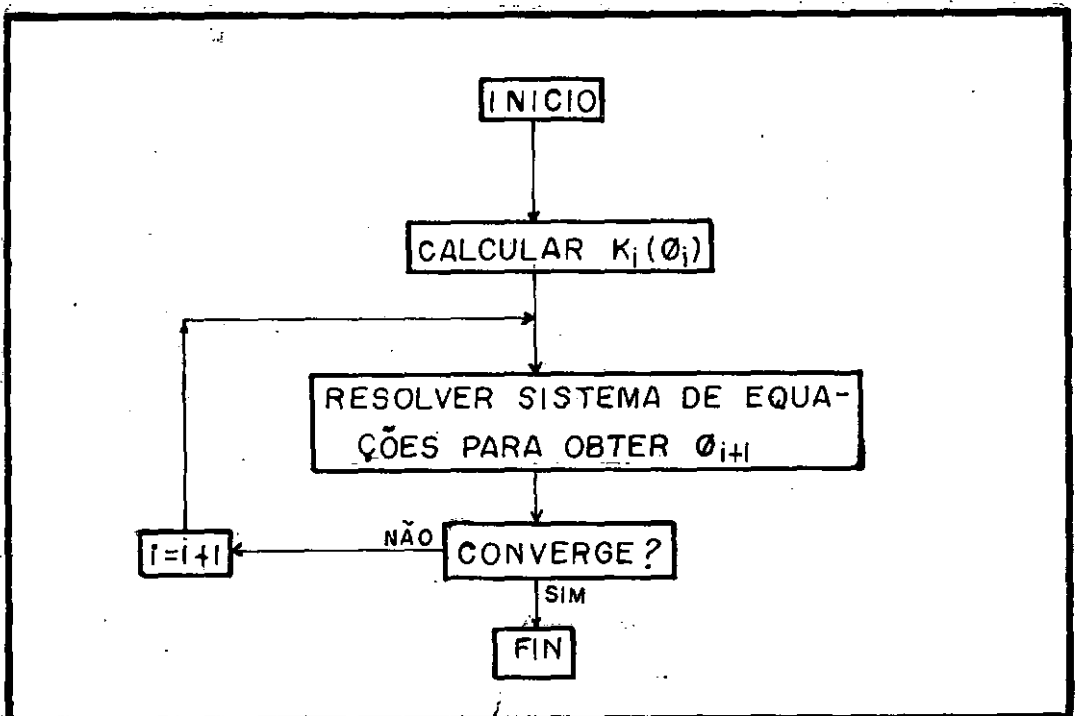


Figura 6.4
Método de Iteração Direta

b) Newton-Raphson

Neste método se utiliza a matriz K tangente para se resolver o sistema de equações, pelo que se trabalha baseado em um desequilíbrio ΔP . O método é o seguinte:

A matriz tangente se constitui com base nos valores correspondentes

$$\begin{bmatrix} K_i^T \end{bmatrix} = \begin{bmatrix} K_i^T (\phi_i) \end{bmatrix} \quad (6.20)$$

donde

$$K_T(\phi) = \frac{dP}{d\phi} \quad (6.21)$$

Com esta matriz e o desequilíbrio ΔP_i que se conhece com relação ao nível de carga total, resolve-se o sistema:

$$\begin{bmatrix} K_i^T \end{bmatrix} \begin{bmatrix} \Delta \phi_i \end{bmatrix} = \begin{bmatrix} \Delta P_i \end{bmatrix} \quad (6.22)$$

Somando o incremento obtido:

$$\phi_i + 1 = \phi_i + \Delta \phi_i \quad (6.23)$$

Com estes valores se constroi a matriz K,

$$\begin{bmatrix} K_i + 1 \end{bmatrix} = \begin{bmatrix} K_i + 1 (\phi_i + 1) \end{bmatrix} \quad (6.24)$$

com o que se obtém um novo nível de cargas

$$\begin{bmatrix} P_i + 1 \end{bmatrix} = \begin{bmatrix} K_i + 1 \end{bmatrix} - \begin{bmatrix} \phi_i + 1 \end{bmatrix} \quad (6.25)$$

Este nível de cargas, tendo em vista que a solução $\begin{bmatrix} \phi_i + 1 \end{bmatrix}$ não é a correta, apresenta uma certa diferença com o nível total de cargas, pelo que se terá um novo ΔP :

$$\begin{bmatrix} \Delta P_i + 1 \end{bmatrix} = \begin{bmatrix} P_i + 1 \end{bmatrix} - \begin{bmatrix} P_i \end{bmatrix} \quad (6.26)$$

Com o que se pode iniciar novamente o processo pela equação (6.19).

A Figura 6.5 mostra um diagrama do método de Newton-Raphson e na Figura 6.6 foi representado o diagrama de fluxo correspondente.

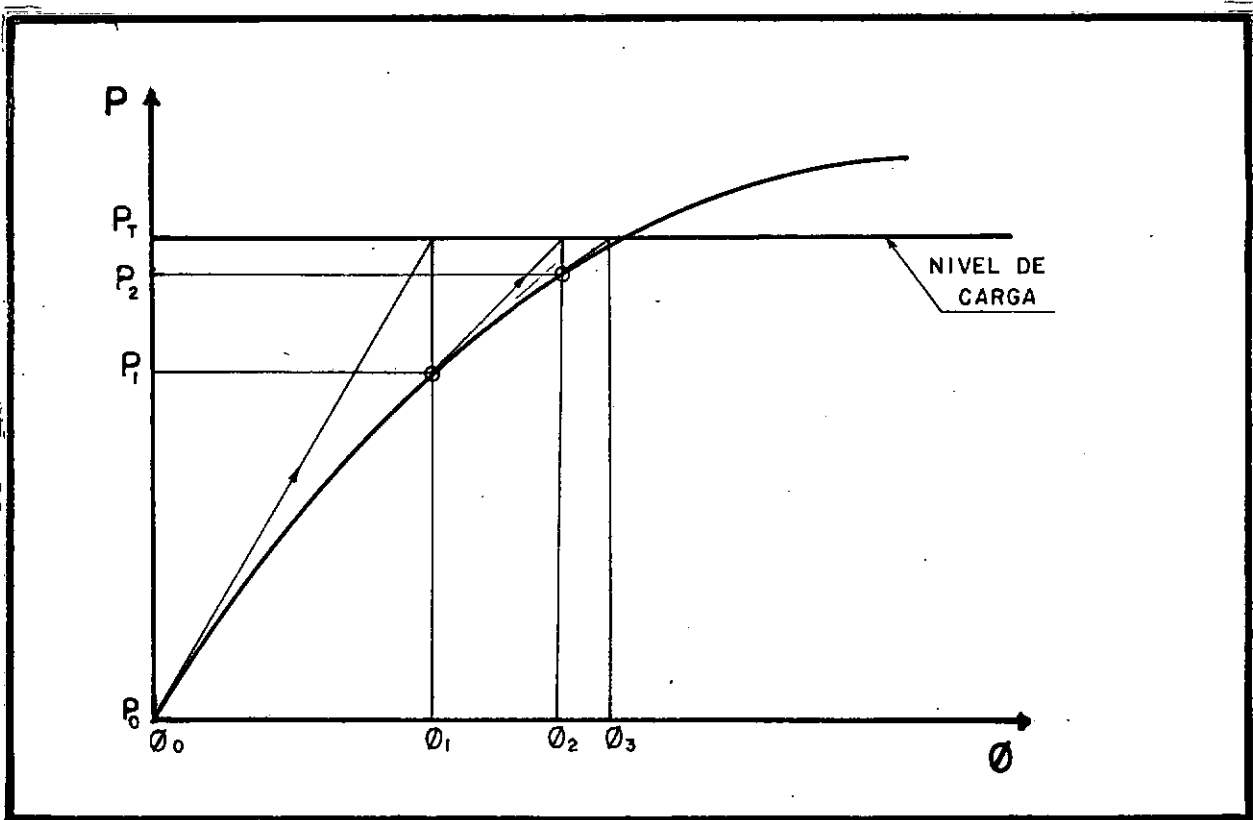


Figura 6.5
Newton - Raphson

O processo de Newton-Raphson é usualmente convergente na proximidade da solução, mas se a suposição inicial não é adequada para a forma da curva, pode novamente ocorrer divergência no processo. Isto geralmente se dá em casos onde a solução se encontra muito próxima de um ponto de inflexão da curva [23] Ver Fig. 6.7.

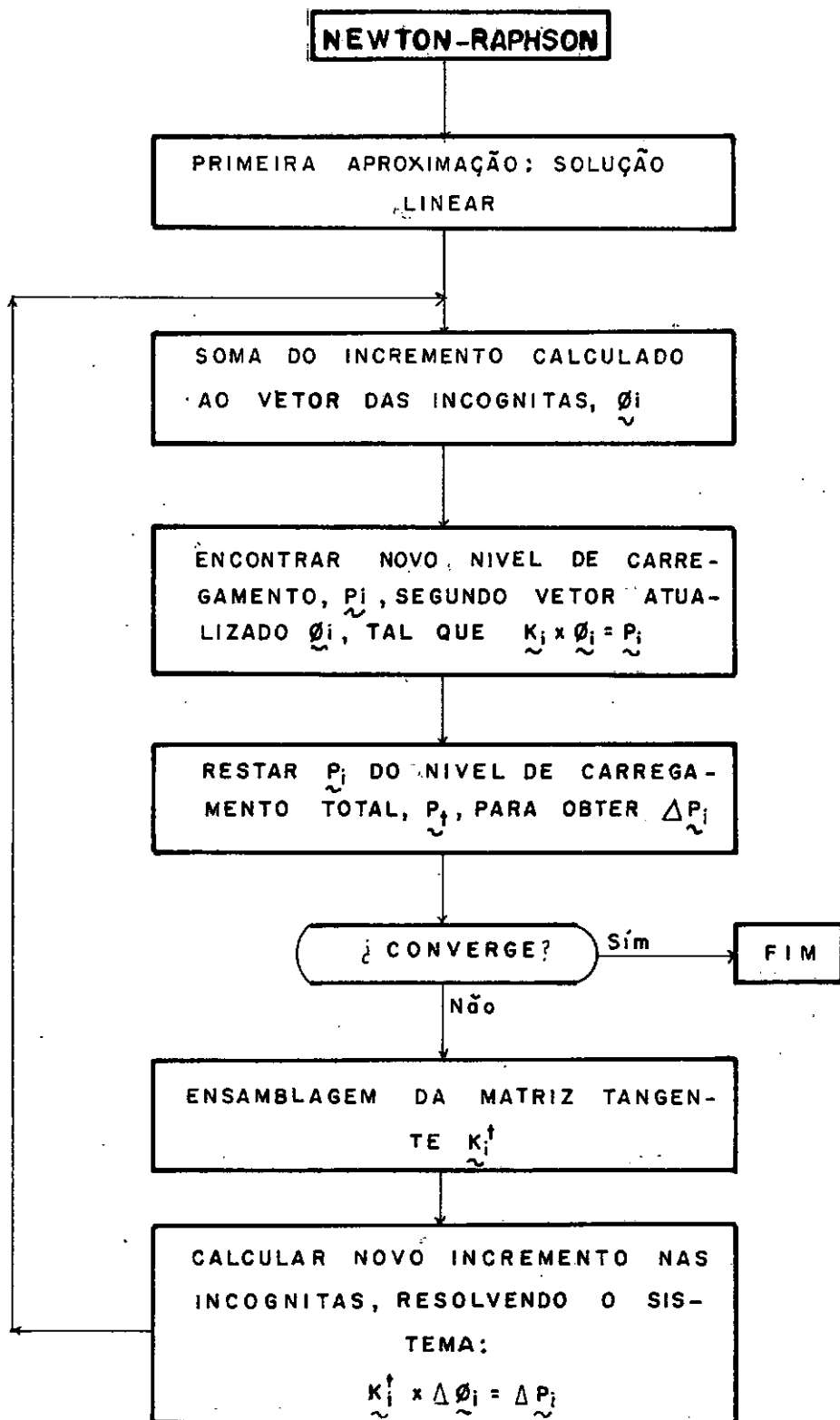


Figura 6.6

Newton-Raphson

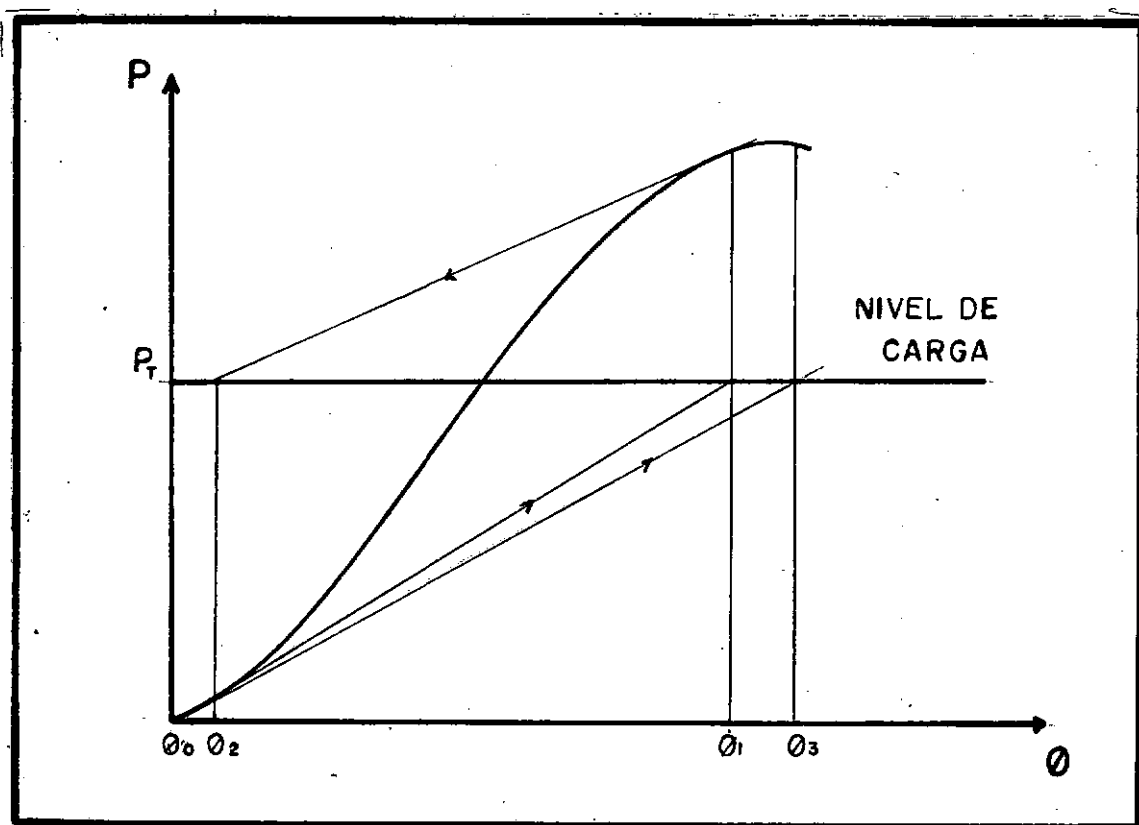


Figura 6.7
Divergência no Newton-Raphson

Cabe mencionar neste ponto que em alguns tipos de problemas a matriz tangente pode resultar não simétrica, razão pela qual este método pode não ser eficiente. Tal é o caso dos problemas não lineares de temperatura, como foram formulados no Capítulo IV. Existem outros problemas de campo não lineares onde a variação dos parâmetros com relação à variável é tal, que permite chegar a uma matriz tangente simétrica; faz-se referência a alguns problemas de fluxo em meios porosos, problemas de campo magnético, etc [18]. Pode-se assegurar que a matriz tangente será simétrica se a matriz K provier de um princípio variacional com base em funcional quadrático, como ocorre na maioria dos problemas de estruturas.

c) Newton-Raphson Modificado

O procedimento anterior pode se tornar menos custoso se se evita a construção de uma matriz K diferente por cada iteração. Neste sentido ODEN [37] introduziu o que veio a cha-

mar-se de Newton-Raphson modificado, o qual consiste em se utilizar a matriz inicial para o resto das iterações, até chegar à convergência. (Fig. 6.8).

É evidente que este procedimento requererá maior número de iterações que o anterior, para alcançar a convergência, mas isto pode ser melhorado se se modificar a matriz K a cada certo número de iterações.

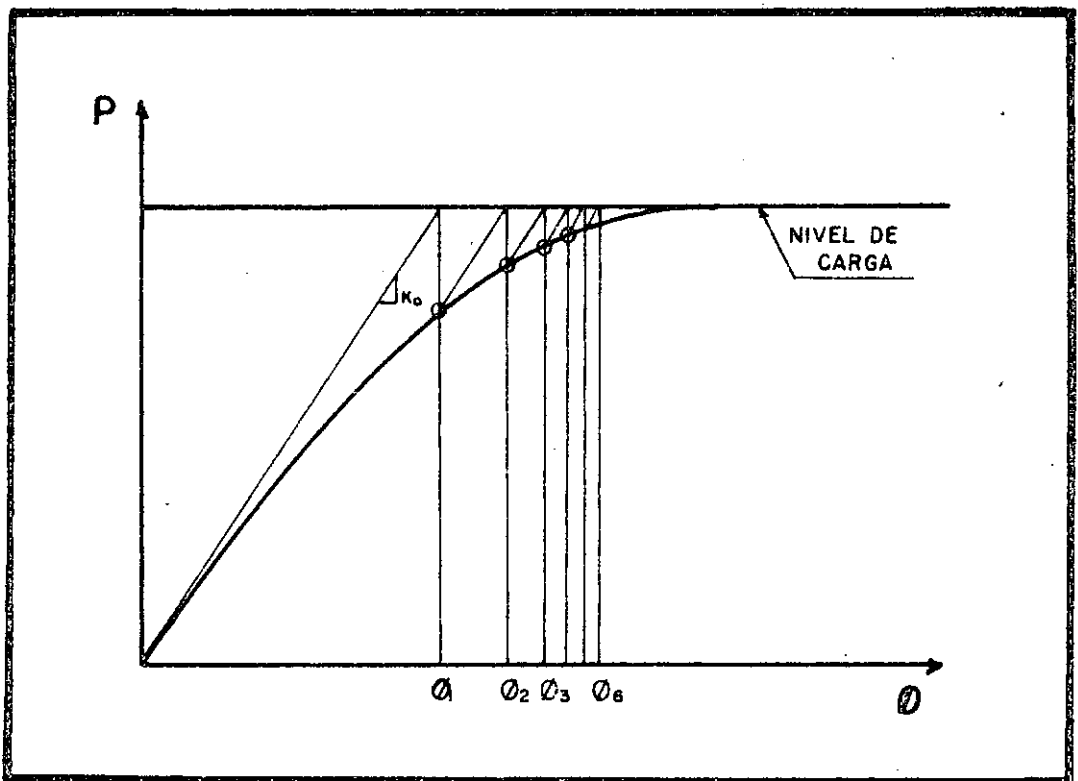


Figura 6.8
Newton-Raphson Modificado

6.3.3 Técnicas Mixtas

Em muitos casos se utilizam combinações dos métodos anteriores, o que, a custa de um esforço computacional maior, produz talvez resultados mais precisos. Na Fig. 6.9 se mostra um tipo de análise incremental, onde para cada incremento se utiliza um método iterativo na solução.

6.3.4 Comentários

mar-se de Newton-Raphson modificado, o qual consiste em se utilizar a matriz inicial para o resto das iterações, até chegar à convergência. (Fig. 6.8).

É evidente que este procedimento requererá maior número de iterações que o anterior, para alcançar a convergência, mas isto pode ser melhorado se se modificar a matriz K a cada certo número de iterações.

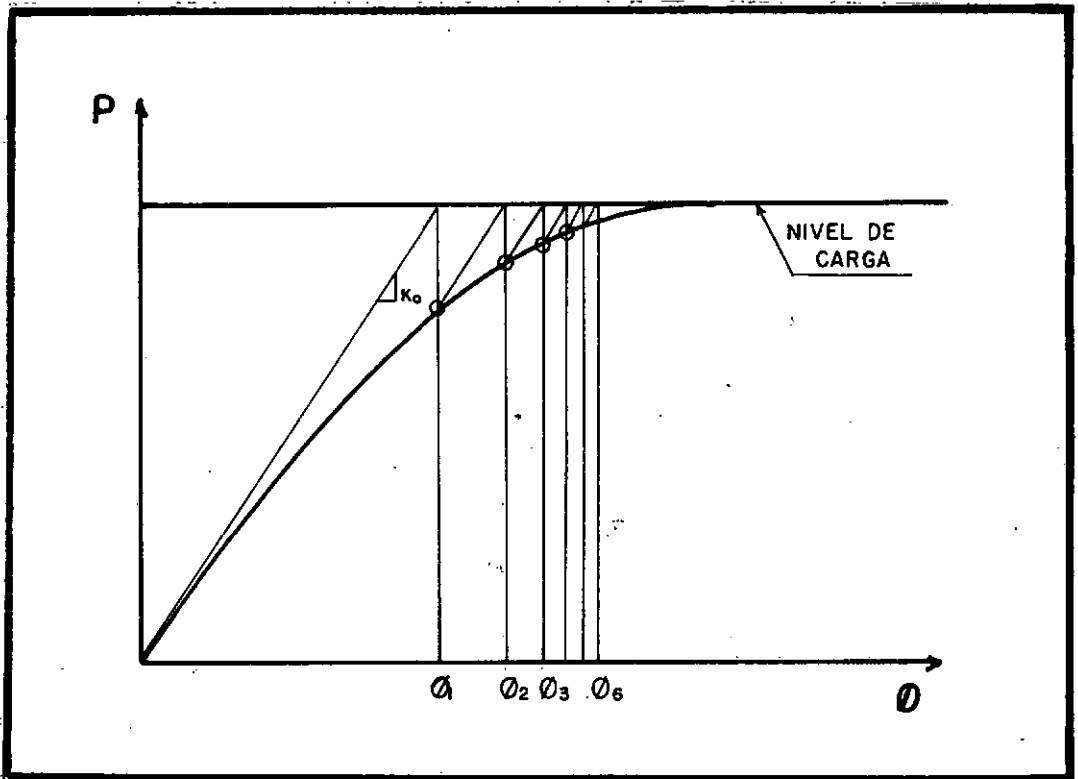


Figura 6.8
Newton-Raphson Modificado

6.3.3 Técnicas Mixtas

Em muitos casos se utilizam combinações dos métodos anteriores, o que, a custa de um esforço computacional maior, produz talvez resultados mais precisos. Na Fig. 6.9 se mostra um tipo de análise incremental, onde para cada incremento se utiliza um método iterativo na solução.

6.3.4 Comentários

As técnicas incrementais também proporcionam resul

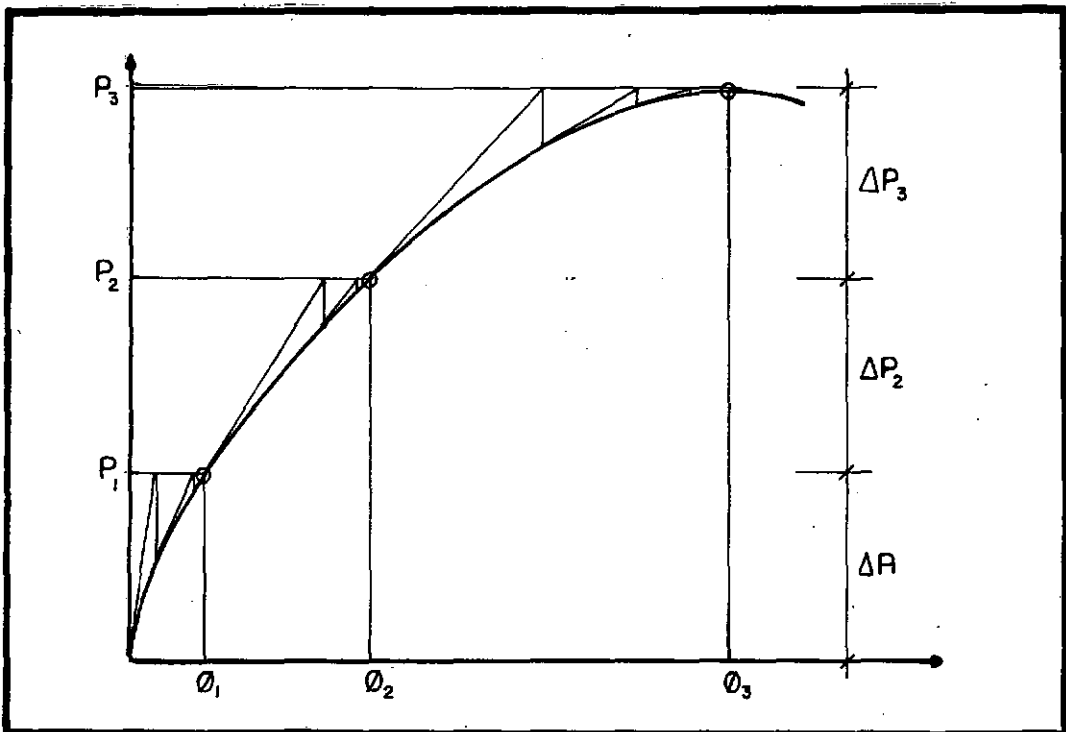


Figura 6.9
Técnicas Mixtas

tados para uma historia de carga, e em parte por este mesmo motivo, são um pouco mais lentas que as técnicas iterativas, nas quais só é necessário estudar-se um caso de carga. Além disso, a menos que se disponha de alguma referência, é difícil de se obter o erro aproximado com relação à solução exata, coisa que se obtém com as soluções do tipo Newton-Raphson. Isto pode ser sanado utilizando-se algum tipo de técnica mixta, o que vem aumentar muito o esforço computacional. A eleição de um intervalo de carga, ΔP , adequado para a obtenção de uma boa aproximação, em alguns casos, também apresenta dificuldades.

De um modo geral, é recomendável o uso das técnicas iterativas, quando se necessita a resposta de um único estado de cargas P , já que estes métodos são mais rápidos que os anteriores. A desvantagem destes métodos é que podem não convergir para uma solução, razão porque é conveniente, como já se mencionou, um conhecimento prévio do comportamento físico do problema e, se é possível, o início das iterações mediante uma solução próxima da desejada.

6.4 - ANÁLISE DE PROBLEMAS DE AUTOVALORES E AUTOVETORES

Em numerosos problemas de engenharia encontram-se formulações que conduzem à resolução de um sistema de autovalores e autovetores; tal é o caso de problemas dinâmicos em estruturas envolvendo vibrações livres, ou alguns dos problemas da equação de campo que derivam da equação de Hemholtz (Cap. IV). Também é necessário efetuar uma análise por autovalores e autovetores como requisito prévio para uma análise dinâmica pelo método de superposição modal, por exemplo.

Em sua forma mais generalizada, o problema de autovalores e autovetores vem representado por:

$$\underline{K} \cdot \underline{\phi}_i = \lambda_i \underline{M} \cdot \underline{\phi}_i \quad (6.27)$$

onde os diferentes valores de λ_i são os "autovalores" do sistema, e os correspondentes vetores $\underline{\phi}_i$ são os "autovetores". As

matrizes \underline{M} e \underline{K} vêm sendo chamadas matrizes de "massa" e de "rigidez", seguindo a tradição estrutural, ainda que em numerosas aplicações em outras áreas, não tenham tais correspondências. Da mesma forma, as matrizes $\underline{\Phi}$, contendo os autovetores, e $\underline{\Lambda}$, contendo os autovalores, serão chamadas matrizes "modais" e "espectrais", respectivamente.

O problema dado por (6.27) também pode apresentar-se em uma forma standard:

$$\underline{K} \underline{\Phi} = \underline{\Phi} \underline{\Lambda} \quad (6.28)$$

No caso em que a matriz \underline{K} seja simétrica e positiva definida, como nas aplicações apresentadas para este trabalho, os autovalores serão números reais e positivos e os autovetores serão independentes, com o que a matriz $\underline{\Phi}$ é não singular.

Para se resolver o problema de autovalores e autovetores existem duas técnicas saponíveis; as técnicas diretas e as técnicas iterativas. Dentre as técnicas diretas se encontram as de Jacobi, Givens, householder e o método QR, mediante os quais se conseguem todos os autovalores e autovetores de um sistema. Dentre as técnicas iterativas, de uso conveniente no caso de grandes sistemas, onde só necessário conhecer alguns dos autovalores e autovetores dos mesmos, encontram-se as de Stodola-Vianello, iteração inversa e iteração por sub-espacos.

[18]

As técnicas implementadas no sistema mostrado na Fig. 3.1 são uma do tipo direto, o método de Jacobi generalizado e a outra do tipo iterativo, o método de iteração por sub-espacos.

6.4.1 - O Método de Jacobi Generalizado

Este método resolve problemas do tipo

$$\underline{K} \underline{\Phi} = \underline{M} \underline{\Phi} \underline{\Lambda} \quad (6.29)$$

sem a necessidade de transformá-los em uma forma standard, já

que atua ao mesmo tempo sobre as matrizes \underline{M} e \underline{K} .

A base do método consiste em, através de iterações, conseguir uma matriz que diagonalize as matrizes \underline{K} e \underline{M} na forma:

$$\underline{\Phi}^T \underline{K} \underline{\Phi} = \underline{\Lambda} \quad (6.30)$$

$$\underline{\Phi}^T \underline{M} \underline{\Phi} = \underline{I} \quad (6.31)$$

a matriz $\underline{\Phi}$ se constroi com base em sucessivas aproximações de uma matriz \underline{R} de rotação, até diagonalizar as matrizes \underline{K} e \underline{M} :

$$\begin{aligned} \underline{K}_2 &= \underline{R}_1^T (\underline{K} \underline{R}_1) \\ \underline{K}_3 &= \underline{R}_2^T \underline{K}_2 \underline{R}_2 \\ &\vdots \\ \underline{K}_{n+1} &= \underline{R}_n^T \underline{K}_n \underline{R}_n \end{aligned} \quad (6.32)$$

e de forma similar

$$\begin{aligned} \underline{M}_2 &= \underline{R}_1^T \underline{M} \underline{R}_1 \\ \underline{M}_3 &= \underline{R}_2^T \underline{M}_2 \underline{R}_2 \\ &\vdots \\ \underline{M}_{n+1} &= \underline{R}_n^T \underline{M}_n \underline{R}_n \end{aligned} \quad (6.33)$$

de tal maneira que, quando R tende ao infinito, as matrizes \underline{M} e \underline{K} tendem a ser diagonais.

O resultado seria para $n \rightarrow \infty$

$$\Lambda = \begin{bmatrix} \frac{K_1}{M_1} & 0 & 0 & \cdot & 0 \\ 0 & \frac{K_2}{M_2} & 0 & \cdot & 0 \\ 0 & 0 & \frac{K_3}{M_3} & \cdot & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \frac{K_m}{M_m} \end{bmatrix} \quad (6.34)$$

$$\Phi = R_1 R_2 R_3 \dots R_n \begin{bmatrix} \frac{1}{\sqrt{M_1}} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \frac{1}{\sqrt{M_2}} & \cdot & \cdot & \cdot & 0 \\ \vdots & \vdots & & \vdots & & \\ 0 & 0 & & & & \frac{1}{\sqrt{M_n}} \end{bmatrix} \quad (6.35)$$

As matrizes R_m de rotação se constroem na forma:

$$R_m = \begin{bmatrix} \text{coluna i} & \text{coluna j} \\ \begin{matrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots \end{matrix} & \begin{matrix} \updownarrow & \updownarrow \\ \alpha & 1 \end{matrix} \\ \begin{matrix} \leftarrow & \leftarrow \\ \text{fila i} & \text{fila j} \end{matrix} \\ \vdots & \vdots \\ \begin{matrix} & & & 1 \end{matrix} \end{bmatrix} \quad (6.37)$$

onde os coeficientes α e γ são selecionados de forma tal que reduzem a zero simultaneamente os elementos K_{ij} e M_{ij} da iteração m .

Resolvendo o sistema de equações resultante desta condição tem-se:

$$\gamma = - \frac{\bar{K}_{ii}^{(m)}}{x} \quad (6.37)$$

$$\alpha = \frac{\bar{K}_{jj}^{(m)}}{x} \quad (6.38)$$

donde:

$$\bar{K}_{ii}^{(m)} = K_{ii}^{(m)} M_{ij}^{(m)} - M_{ii}^{(m)} K_{ij}^{(m)} \quad (6.39)$$

$$\bar{K}_{jj}^{(m)} = K_{jj}^{(m)} M_{ij}^{(m)} - M_{jj}^{(m)} K_{ij}^{(m)} \quad (6.40)$$

$$\bar{K}^{(m)} = K_{ii}^{(m)} M_{jj}^{(m)} - K_{jj}^{(m)} M_{ii}^{(m)} \quad (6.41)$$

$$x = \frac{\bar{K}^{(m)}}{2} + \text{sign}(K^{(m)}) \sqrt{\left(\frac{\bar{K}^{(m)}}{2}\right)^2 + \bar{K}_{ii}^{(m)} \bar{K}_{jj}^{(m)}} \quad (6.42)$$

Se o valor x é zero, quer dizer que se apresenta um caso trivial de:

$$\frac{K_{ii}^{(m)}}{m_{ii}^{(m)}} = \frac{K_{jj}^{(m)}}{m_{ii}^{(m)}} = \frac{K_{ij}^{(m)}}{m_{ij}^{(m)}} \quad (6.43)$$

e se utiliza:

$$\alpha = 0 \quad (6.44)$$

$$\gamma = - \frac{K_{ij}^{(m)}}{K_{jj}^{(m)}} \quad (6.45)$$

Para que o algoritmo trabalhe é necessário que o determinante das matrizes \underline{R}_m seja diferente de zero, o que acontece se as matrizes \underline{M} e \underline{K} são positivas definidas.

Ao fazer de zero um elemento das matrizes \underline{M} e \underline{K} é possível que elementos anteriormente zerados adquiram valores diferentes, o que torna necessário varrer todos os elementos das matrizes em cada iteração. Um dos métodos para se efetuar isto é o de se indagar qual é o maior dos elementos fora da diagonal nas matrizes, o que consome muito tempo para investigar cada um de per si. Por isto, é preferível checar todos os elementos sequencialmente, contra erro ou tolerância [24] que varia segundo o número de varreduras que se haja efetuado. Esta verificação é feita através de:

$$\left[\frac{(K_{ij}^{(m)})^2}{K_{ii}^{(m)} K_{jj}^{(m)}} \right]^{1/2} < 10^{-2m} \quad (6.46)$$

$$\left[\frac{(M_{ij}^{(m)})^2}{M_{ii}^{(m)} M_{jj}^{(m)}} \right]^{1/2} < 10^{-2m} \quad (6.47)$$

onde m é o número de vezes em que se varreu a matriz.

Quando se consegue que um dos fatores de acoplamento, dados pelas fórmulas anteriores, seja maior que a tolerância, procede-se o zeramento do elemento ij correspondente em ambas as matrizes.

A convergência se alcança quando:

$$\frac{|\lambda_i^{(m+1)} - \lambda_i^{(m)}|}{\lambda_i^{(m+1)}} \leq \text{TOL} \quad i=1, \dots, n \quad (6.48)$$

onde os λ_i são os autovalores nas iterações m e $m+1$.

E, simultaneamente

$$\left[\frac{(K_{ij}^{(m+1)})^2}{K_{ii}^{(m+1)} K_{jj}^{(m+1)}} \right]^{1/2} \leq \text{TOL} \quad (6.49)$$

$$\left[\frac{(M_{ij}^{(m+1)})^2}{M_{ii}^{(m+1)} M_{jj}^{(m+1)}} \right]^{1/2} \leq \text{TOL} \quad (6.50)$$

onde TOL é a tolerância permitida.

O diagrama de fluxo correspondente à implementação deste método se encontra na Fig.(6.10).

Este método é recomendável quando se requer o conhecimento de todos os autovalores e autovetores de um sistema. Para sistemas muito grandes é dispendioso e é pouco empregado; por este motivo sua implementação se realizou em memória direta, sem utilização do disco. Este módulo é usado quando se vai realizar uma análise por sub-espacos.

6.4.2 - O Método de Iteração por Sub-Espacos

Em problemas de elementos finitos geralmente se requer somente os menores autovalores e autovetores de um sistema. Por este motivo, é muito mais barato utilizar um método que permita calcular só autovalores e autovetores requeridos, sobretudo quando as matrizes são grandes e requerem o uso de memória externa de suporte. Nestes métodos se encontram o de "busca da determinante" e o "método de iteração por sub-espacos". Destes o último é o que foi implementado para este trabalho, juntamente com rotinas que realizam uma checagem dos autovalores calculados pelo método da sequência de Sturm [26].

A implementação se realizou baseado nas matrizes \underline{K} e \underline{M} positivas definidas, sendo esta última consistente. Aproveita-se como em outros casos anteriores, o esquema em banda e

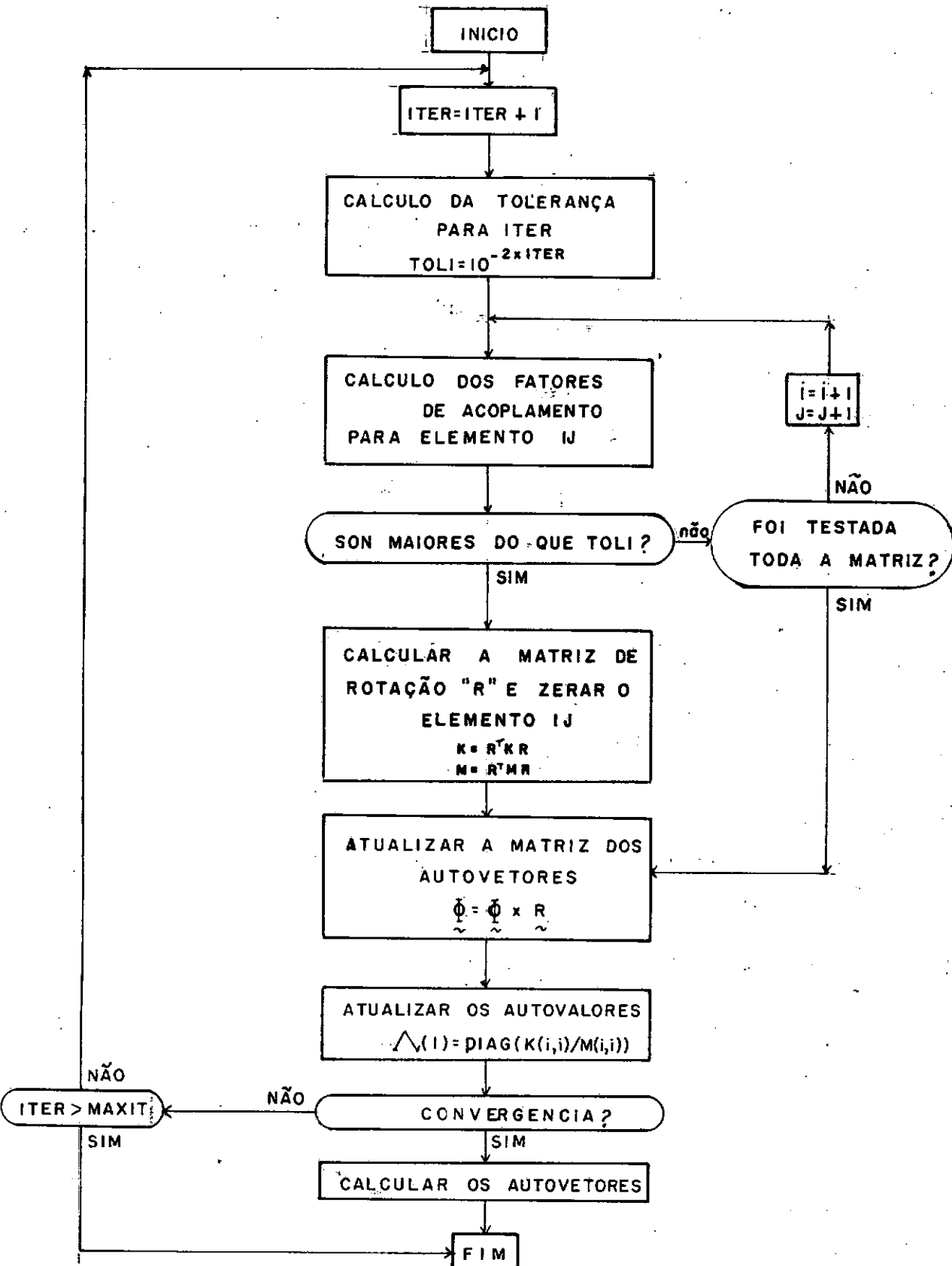


Figura 6.10
Método de Jacobi

simétrico para o armazenamento das matrizes.

A capacidade de armazenamento é a que proporciona o disco como memória suporte, o qual permite resolver problemas de tamanho muito grande. O manejo de informação se realiza com base na função do Sapo (Cp.III). A matriz \underline{M} se armazena em seguida da matriz \underline{K} , por filas no arranjo unidimensional do disco, e os vetores de iteração de forma adjacente à mesma matriz, como parte de suas filas, o que agiliza a solução do sistema (Ver seção 6.2). Um esquema geral se mostra na Figura(6.11).

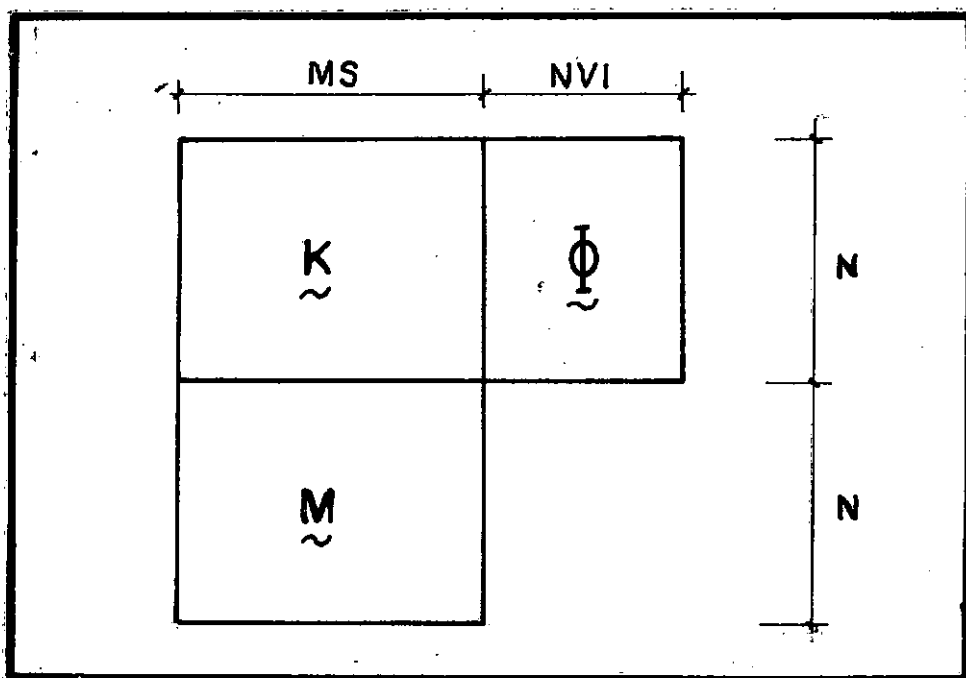


Figura 6.11

Esquema Geral de Armazenamento no Disco

Os vetores auxiliares necessários para levar a cabo o processo também são armazenados em disco e manejados através da função Sapo.

6.4.2.1 - Descrição Geral do Método: É um método de iteração inversa simultâneo combinado com uma análise de Rayleig Ritz, com base em um conjunto de vetores de iteração cujo núme-

mero é maior que o número de autovalores e autovetores que se deseja obter [26].

O objetivo do método é conseguir os p menores autovalores do sistema.

$$\underline{K} \underline{\Phi} = \underline{M} \underline{\Phi} \underline{\Lambda} \quad (6.51)$$

onde $\underline{\Phi}$ é a matriz dos p autovetores e $\underline{\Lambda}$ a dos p autovalores.

Uma das vantagens deste método é de que as iterações se realizam baseadas em espaços vetoriais definidos por p vetores independentes entre si, e não em base de vetores individuais, como no algoritmo de iteração inversa. Isto faz com que a convergência seja mais rápida, já que, se em algum momento os vetores de iteração são combinações lineares dos autovetores requeridos, o espaço gerado é o espaço vetorial de convergência, e o processo de iteração termina. O método de iteração por sub-espaços cria para cada iteração uma base ortogonal de vetores que geram o espaço em curso, razão pela qual se conserva a estabilidade da iteração ao não permitir que os vetores que geram o espaço vão fazendo paralelos entre si.

O processo de iteração se realiza como segue: [26]

$$\underline{K} \bar{\underline{X}}_{k+1} = \underline{M} \underline{X}_k \quad (6.52)$$

os operadores K e M se projetam sobre o espaço gerado por $\bar{\underline{X}}_{k+1}$:

$$\underline{K}_{k+1} = \bar{\underline{X}}_{k+1}^T \underline{K} \bar{\underline{X}}_{k+1} \quad (6.53)$$

$$\underline{M}_{k+1} = \bar{\underline{X}}_{k+1}^T \underline{M} \bar{\underline{X}}_{k+1} \quad (6.54)$$

Resolve-se o sistema resultante neste espaço, por algum método de solução direto:

$$\underline{K}_{k+1} \bar{\underline{Q}}_{k+1} = \underline{M}_{k+1} \bar{\underline{Q}}_{k+1} \underline{\Lambda} \quad (6.55)$$

e se consegue uma aproximação melhorada para os autovetores:

$$\underline{X}_{k+1} = \underline{X}_{k+1} \underline{Q}_{k+1} \quad (6.56)$$

Se os vetores iniciais não são ortogonais a algum dos vetores requeridos, o processo repetido das fórmulas (6.52)a (6.56) conduz a:

$$\begin{aligned} \underline{\Lambda}_{k+1} &\rightarrow \underline{\Lambda} \\ \underline{X}_{k+1} &\rightarrow \Phi \end{aligned} \quad (6.57)$$

Os autovalores devem ser ordenados de menor a maior, o mesmo acontecendo com os autovetores. Desta maneira os autovalores menores convergem mais rápido que os seguintes, com uma média de convergência igual a:

$$\text{CONVERGÊNCIA} = \frac{\lambda_i}{\lambda_{p+1}} \quad (6.58)$$

A convergência se fará mais rápida quando se escolhe um número de vetores de iteração maior que o número "p" de vetores desejados. No programa implementado se utiliza, por "default", um número "q" de vetores, tal que:

$$q = \min \{2p, p+8\} \quad (6.59)$$

A eficácia do método depende muito dos vetores escolhidos para a 1ª iteração. Um algoritmo eficiente de eleição destes vetores é o seguinte:

- a) A primeira coluna da matriz $\underline{M} \cdot \underline{X}_1$ na fórmula (6.52) é a diagonal de \underline{M} .
- b) As restantes colunas de $\underline{M} \cdot \underline{X}_1$ são vetores unitários, com a unidade colocada nas filas correspondentes aos maiores valores de M_{ii}/K_{ii} .

O primeiro vetor é colocado com o fim de excitar todos os modos de vibração, assegurando que o conjunto de vetores iniciais \underline{X}_1 não seja ortogonal à base desejada. A solução sugerida por (b) se realiza na suposição de que esteja mais ou menos próxima à real, enquanto que a banda se faz menor. Efetivamente constitui a solução, quando K e M são matrizes diagonais, que é o caso extremo.

A convergência das iterações se mede com relação à tolerância dada:

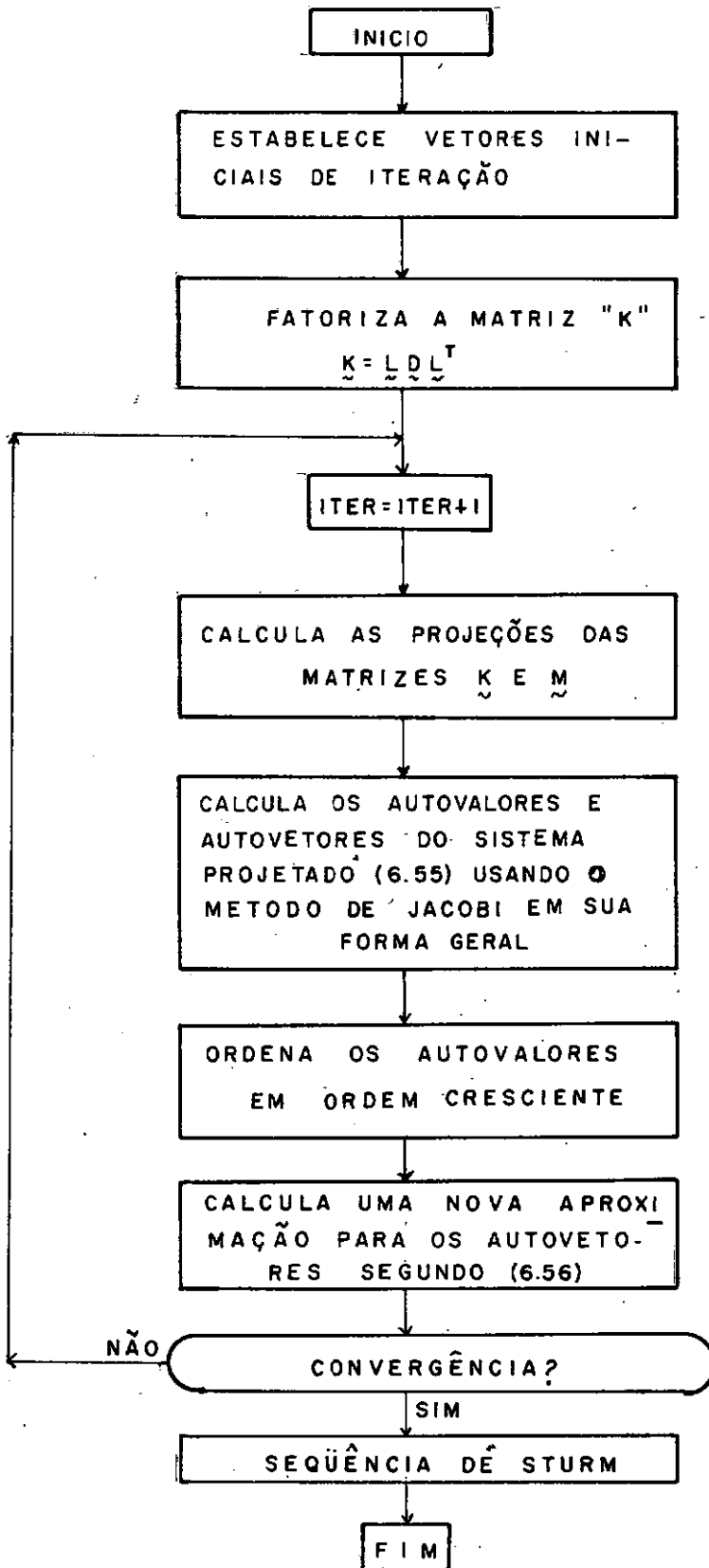


Figura 6.12
Iteração por Sub-espços

$$\frac{|\lambda_i^{(n+1)} - \lambda_i^{(k)}|}{\lambda_i^{(k+1)}} \leq \text{TOL} \quad i = 1, \dots, p \quad (6.60)$$

sobre o número de autovalores desejados.

Se se deseja uma precisão de 2s dígitos nos autovalores, a tolerância deve ser fixada em 10^{-2s} . A precisão dos autovalores será de cerca de um mínimo de s dígitos. Em geral, os autovalores convergem com maior precisão.

Um diagrama de fluxo do método implementado é mostrado na Figura (6.12).

6.4.2.2 - Revisão dos Resultados pelo Método da Sequência de Sturm: Uma vez alcançada a convergência segundo a fórmula (6.60), pode-se revisar se em realidade foram calculados os menores autovalores e autovetores.

Se se realiza um "shift" μ sobre o sistema dado por (6.51), donde μ é somente um pouco maior que λ_p , a fatorização de Gauss, de tal maneira que

$$(\underline{K} - \mu \underline{M}) = \underline{L} \underline{D} \underline{L}^T \quad (6.61)$$

dará como resultado um número de elementos negativos em \underline{D} igual ao número de autovalores menores que μ , isto é, a p.

Seguindo novamente a referência [] os bordos dos autovalores considerados para realizar o "shift" são:

$$0.99 \lambda_i^{(\ell+1)} < \lambda_i < 1.01 \lambda_i^{(\ell+1)} \quad (6.62)$$

CAPÍTULO VII

UTILIZAÇÃO DO SISTEMA

7.1 INTRODUÇÃO

Em capítulos anteriores descreveu-se de uma forma geral, o funcionamento do sistema e a maneira em que está estruturado; referiu-se aos módulos de análises que se encontram em funcionamento e a alguns tipos de problemas que podem ser resolvidos por meio deles, assim como a facilidades no que diz respeito à entrada de dados e ao manejo da informação por meio do uso de memória de suporte.

Neste capítulo são mostrados alguns exemplos de aplicação do sistema, dando idéia de seus alcances e limitações em problemas de análises linear e não-linear e problemas de autovalores e autovetores. Por último são incluídos alguns exemplos das entradas de dados utilizando-se os comandos da linguagem orientada.

7.2 EXEMPLOS DE APLICAÇÃO PARA ANÁLISE LINEAR

7.2.1 - Torção de Saint Venant: As características gerais deste tipo foram dadas na seção (4.3.2). Neste caso foi estudado o comportamento de uma barra prismática de seção quadrada submetida a momento torsor, como mostra a Fig. 7.1.

A seção transversal da barra forma um quadrado de 8 x 8, para o qual se dotou a malha de elementos finitos triangulares mostrada na Fig. 7.2, a qual aproveita a simetria do problema. Na realidade, representa-se somente uma quarta parte da seção com um total de 85 nós e 136 elementos.

As condições de contorno dadas correspondem a $\bar{\Phi} = 0$ (função esforço) sobre os contornos externos, $\frac{d\bar{\Phi}}{dx} = 0$ sobre o eixo de simetria vertical e $\frac{d\bar{\Phi}}{dy} = 0$ sobre o eixo de simetria horizontal.

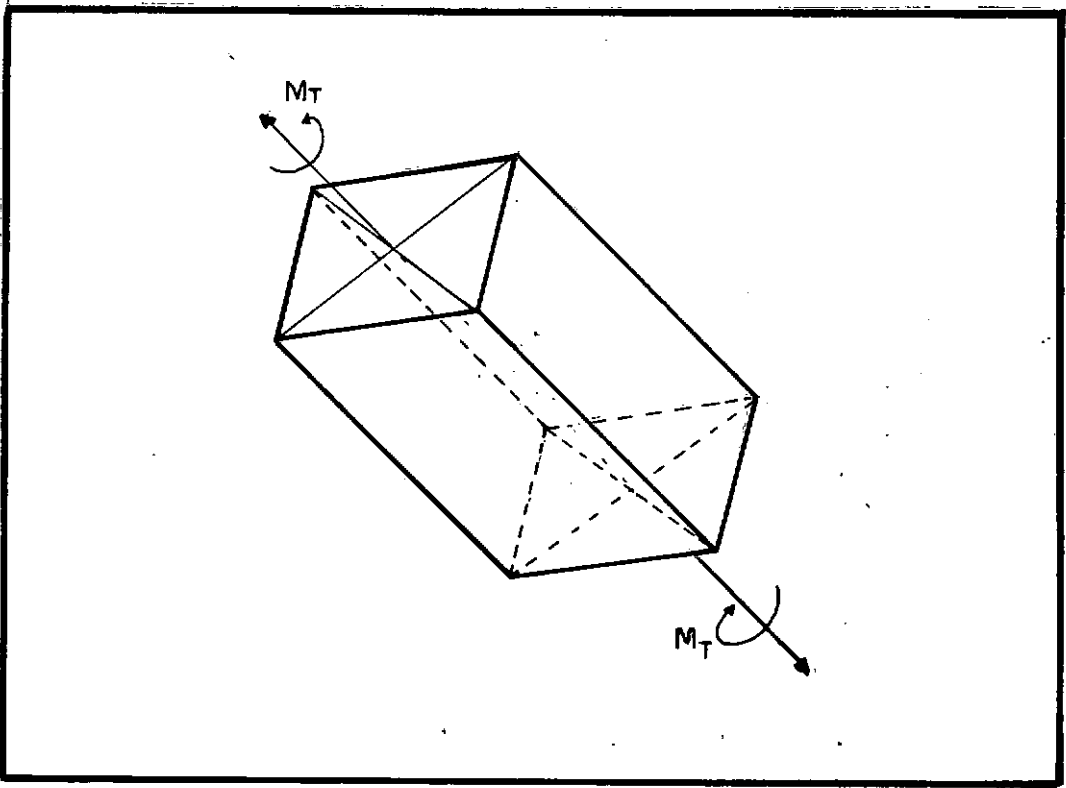


Figura 7.1
Barra Prismática Submetida a Torção

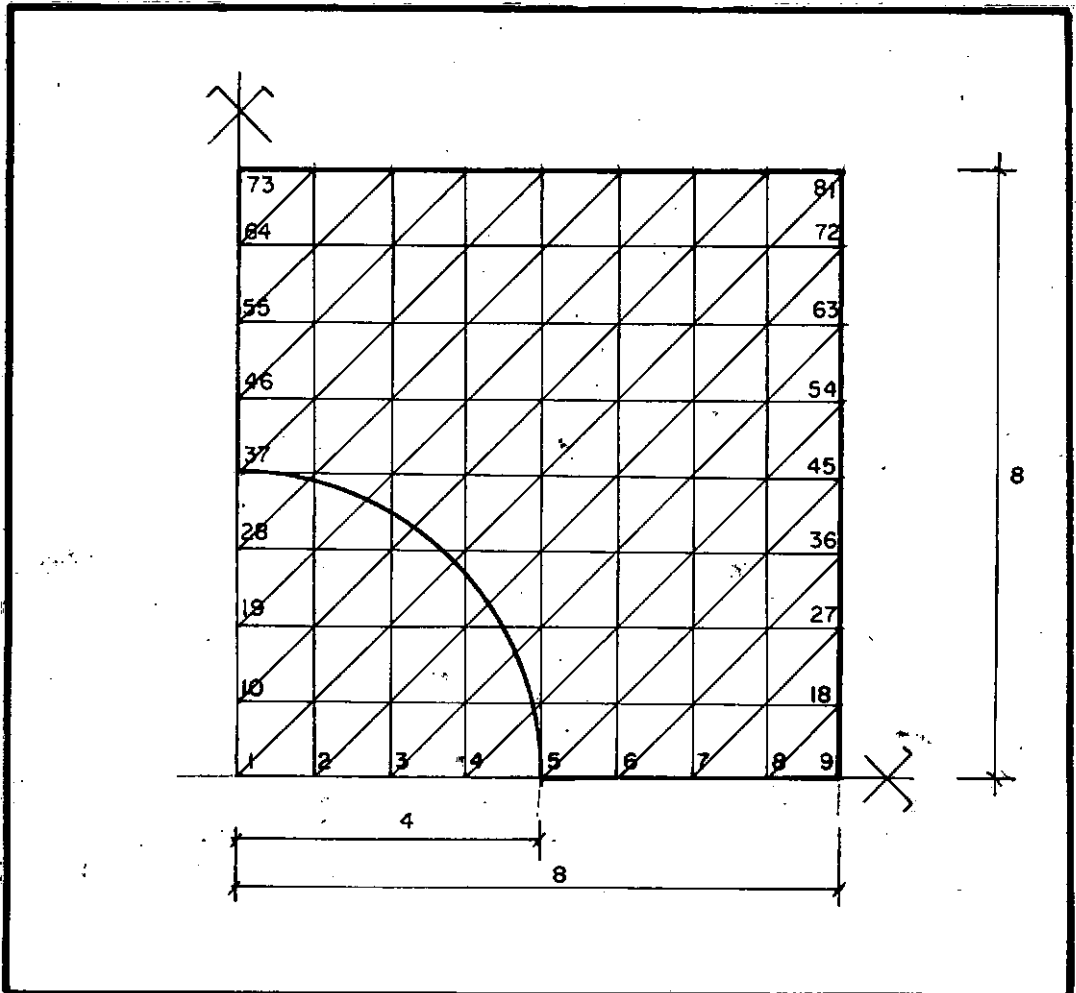


Figura 7.2
Discretização do Modelo

Os resultados encontrados pelo programa foram comparados com a solução exata [33], que proporciona as seguintes fórmulas.

$$\phi = \frac{32G\theta a^2}{\pi^3} \sum_{n=1,3,\dots}^{\infty} \frac{1}{n^3} (-1)^{\frac{n-1}{2}} \left[1 - \frac{\cosh(\frac{n\pi y}{2a})}{\cosh(\frac{n\pi b}{2a})} \cos \frac{n\pi x}{2a} \right] \quad (7.1)$$

$$\tau_{yz} = - \frac{\partial \phi}{\partial x} = \frac{16'G\theta a}{\pi^2} \sum_{n=1,3,\dots}^{\infty} \frac{1}{n^2} (-1)^{\frac{n-1}{2}} \left[1 - \frac{\cosh(\frac{n\pi y}{2a})}{\cosh(\frac{n\pi b}{2a})} \sin \frac{n\pi x}{2a} \right] \quad (7.2)$$

onde b e a são respectivamente a metade de dois lados comprido e curto da seção.

O valor do corte máximo se encontra no centro do lado mais largo do contorno e vem dado por

$$\tau_{\max} = 1,351 G\theta a \quad (7.3)$$

Para um valor de G unitário, a resposta do programa é: (nó 9)

$$\tau_{\max} = 4,93272 \theta \quad (7.4)$$

que comparado com a solução dada por (7.3) dá um erro de 9% aproximadamente.

O erro é muito menor nos valores da função, como mostra a Tabela 7.1

O mesmo problema foi estudado com a presença na barra de um orifício central, tal como mostra a Fig. 7.2. Este foi modelado através de elementos "fictícios" com um módulo de elasticidade nulo, para evitar dar a condição de contorno correspondente a ϕ no contorno interno.

Nº	FÓRMULA (7.1) n = 7	SOLUÇÃO APROXIMADA	ERRO %
38	7. 324046	7. 3638	0.54
55	4. 091850	4. 0574	0.84
26	4. 291156	4. 2658	0.59
69	2. 434670	2. 4320	0.11
67	-6.28.10 ⁻⁹	0. 0000	0.00
75	0. 776745	0. 7905	1,77

Tabela 7.1
Valores de ϕ em Função de θ

Como era de se esperar, a rigidez torsional na barra sólida, dada pela fórmula 7.5 é maior que na segunda, como mostra a Tabela 7.2.

$$J = \frac{2}{G\theta} \iint_A \phi \, dA \quad (7.5)$$

CASO DE SECÇÃO	RIGIDEZ TORSIONAL
Cheia	143.36
Ôca	134.90

Tabela 7.2

O valor da função esforço no bordo circular interno é constante e igual a:

$$\phi = 7.364 \cdot \theta$$

onde θ é a rotação da seção considerada.

7.2.2 - Fluxo Potencial: Para este tipo de problemas foi considerado o caso de um fluxo laminar em regime permanente, tal como se especificou na seção (4.3.3). Procura-se conhecer a distribuição das linhas de corrente em volta de um cilindro entre duas placas paralelas, tal como mostra a Fig. 7.3.

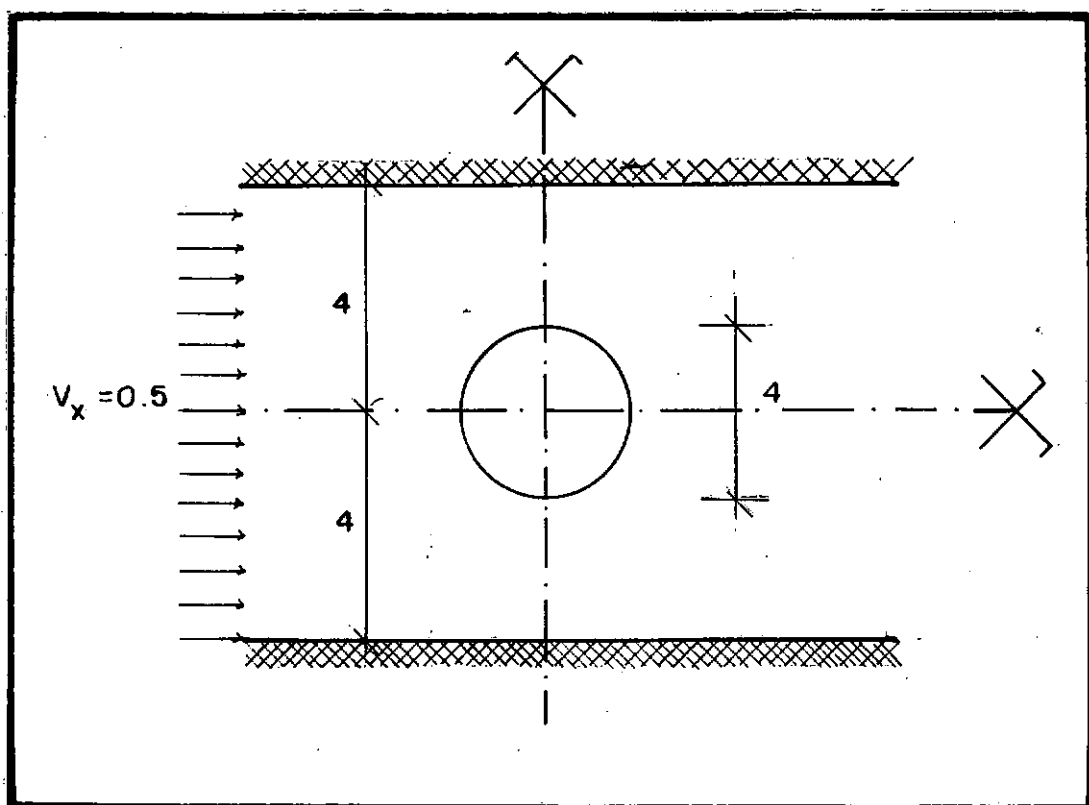


Figura 7.3
Fluxo em Volta de um Cilindro

Aproveitou-se a simetria representando somente uma quarta parte do problema, utilizando para isto uma malha de 108 nós e 176 elementos. O fluxo foi suposto circulando com velocidade constante em uma região fora do cilindro. Esta suposição e a consideração de que as placas sejam totalmente impermeáveis, definem as condições de contorno da função ψ que são mostradas na Fig. 7.4

A velocidade resultante para os nós sobre o eixo vertical de simetria foi de um valor médio igual a 1, tal como mostram os resultados na Tabela 7.3 e a Fig. 7.5. Isto confir

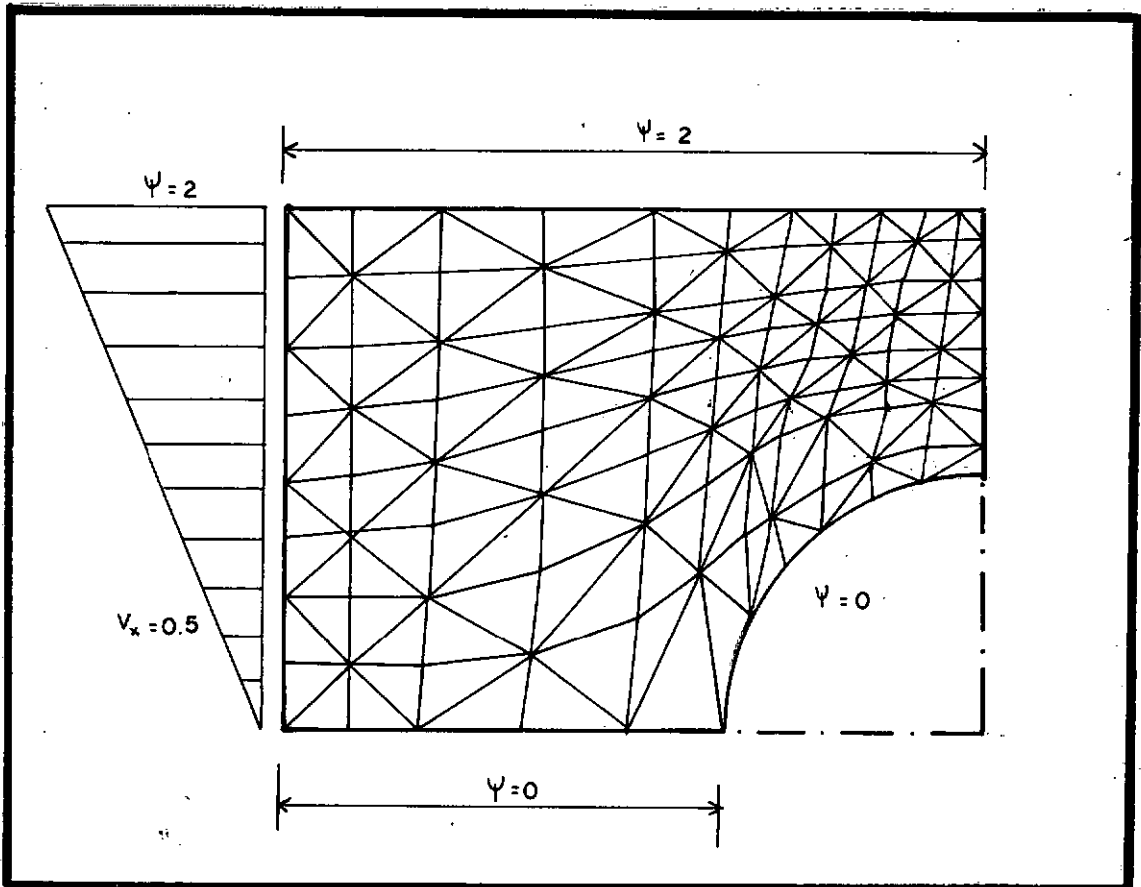


Figura 7.4
Discretização do Modelo
e Condições de Contorno

ma o princípio de continuidade já que a área nesta seção é a metade da da seção esquerda, o que faz com que a velocidade aumente em dobro.

Como é evidente, o erro de aproximação é maior nas velocidades (derivadas da função) que no próprio valor da função, como se observa nos resultados.

ELEMENTO	V_x
21	1.33
22	1.33
41	1.08
42	1.05
43	1.07
44	1.20
45	1.04
65	1.012
66	1.007
88	0.967
110	0.94
131	1.012
132	1.003
154	0.999
175	0.88
176	0.90

Tabela 7.3

Valores da Velocidade
no Eixo Vertical de Simetria

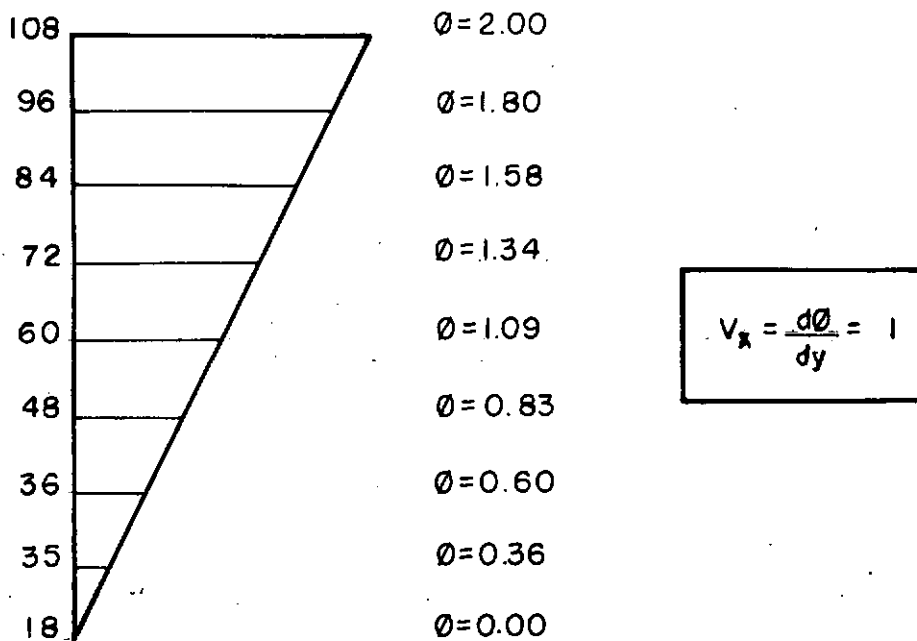


Figura 7.5

Valores de ϕ no eixo Vertical

7.3 EXEMPLOS DE APLICAÇÃO PARA ANÁLISE NÃO LINEAR

7.3.1 - Distribuição de Temperaturas em Placa Quadrada: Considerou-se o caso de uma placa quadrada submetida a diferenças de temperatura em seus bordos e com um coeficiente de condutividade térmica que depende dos valores da temperatura em cada ponto, segundo a fórmula (4.11):

$$K(T) = K_0(1 + \beta T)$$

onde K_0 e β foram tomados como valores unitários para o exemplo. (Ver seção 4.3.1)

Empregou-se uma malha de 32 elementos e 25 nós, e foram especificadas as condições de contorno que indica a Fig. 7.6.

O procedimento utilizado foi o de iteração direta

com a matriz secante, tal como foi explicado na seção 6.3.2, já que para este tipo de problema a matriz tangente resulta não simétrica. O processo convergiu em 6 iterações para uma precisão de 0,00001, o que é bastante aceitável. Os resultados se anexam na Tabela 7.4. Note-se que há um eixo de simetria horizontal na malha

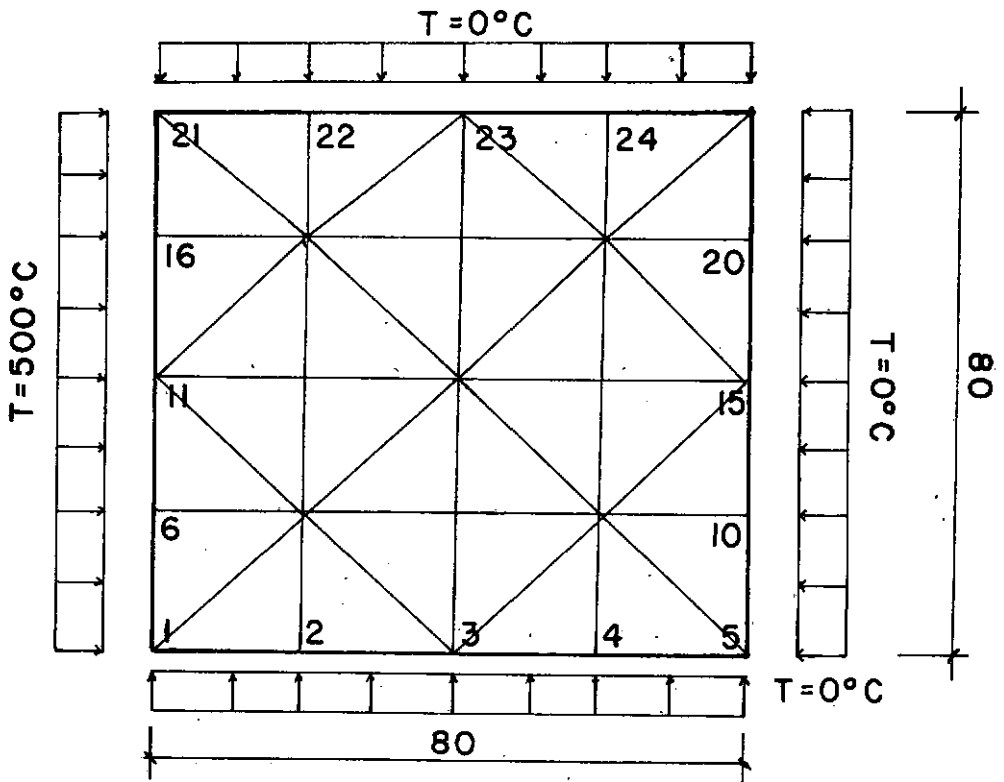


Figura 7.6
Placa Submetida a Mudanças
de Temperaturas

Nº	TEMPERATURA	
	$\beta \cong 0$. SOLUÇÃO LINEAR	$\beta = 1$.
7	214, 2857	311, 8943
8	93, 7500	206, 8169
9	35, 7143	132, 9980
12	263, 3929	352, 7246
13	125, 0000	237, 2823
14	49, 1071	147, 0746
17	214, 2857	311, 8943
18	93, 7500	206, 8169
19	35, 7143	132, 9980

Tabela 7.4

Resultados de Temperatura

7.3.2 - Problema Unidimensional de Distribuição de Temperaturas

Considera-se o caso de uma placa infinita submetida a diferenças de temperatura em seus bordos, como mostra a Fig. 4.1. As condições neste caso conduzem a um problema unidimensional de transmissão de calor resolvido de forma exata pela equação (4.12):

$$\frac{T(x) - T_1}{T_2 - T_1} = \frac{x}{L} + \frac{\beta}{2} \left[(T_2 + T_1) \frac{x}{L} + \frac{T_1^2 - T(x)^2}{T_2 - T_1} \right]$$

O exemplo resolvido neste caso corresponde à Fig. 7.7. Observa-se que a malha utilizada é a mesma que para o exemplo anterior, mas as condições de contorno variam, especificando-se fluxo de calor nulo para os bordos superior e inferior da mesma, condição que faz o problema unidimensional.

Para o problema representado na Fig. 7.7 a resposta do programa é comparada com a solução dada por (4.12) na tabela 7.5. Estes resultados correspondem ao valor $\beta = 1$, e foram obtidos para uma precisão de 0.00001 em 6 iterações, utilizando o método de iteração direta com a matriz secante.

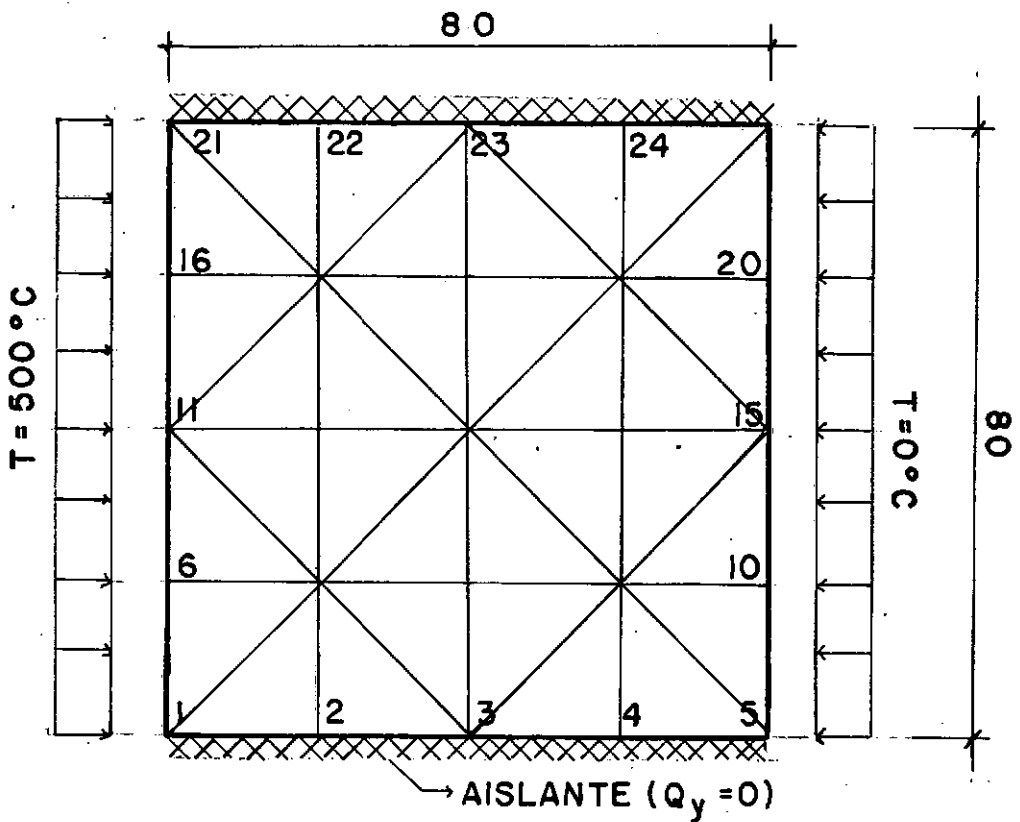


Figura. 7.7
Problema Unidimensional
de Temperatura

A Tabela 7.5 mostra que, para uma malha pouco refinada como a que se utilizou, os resultados do programa são bastante satisfatórios, oferecendo erros menores que 0,2%. Assim também aconteceu com problemas semelhantes ocorrido com diferentes condições geométricas e outras temperaturas de contorno, e a convergência nunca abrangeu mais de seis ou sete iterações.

Nº	VALOR PROGRAMA	VALOR (4.12)	ERRO
11	500.	500.	0%
12	432,9274	432,8790	0,01%
13	353,2782	353,2612	0,005%
14	249,2322	249,5015	0,107%
15	0.	0.	0%

Tabela 7.5
Comparação de Resultados

A Fig. 7.8 mostra o gráfico de valores de temperatura para o exemplo exposto, com diferentes valores de β .

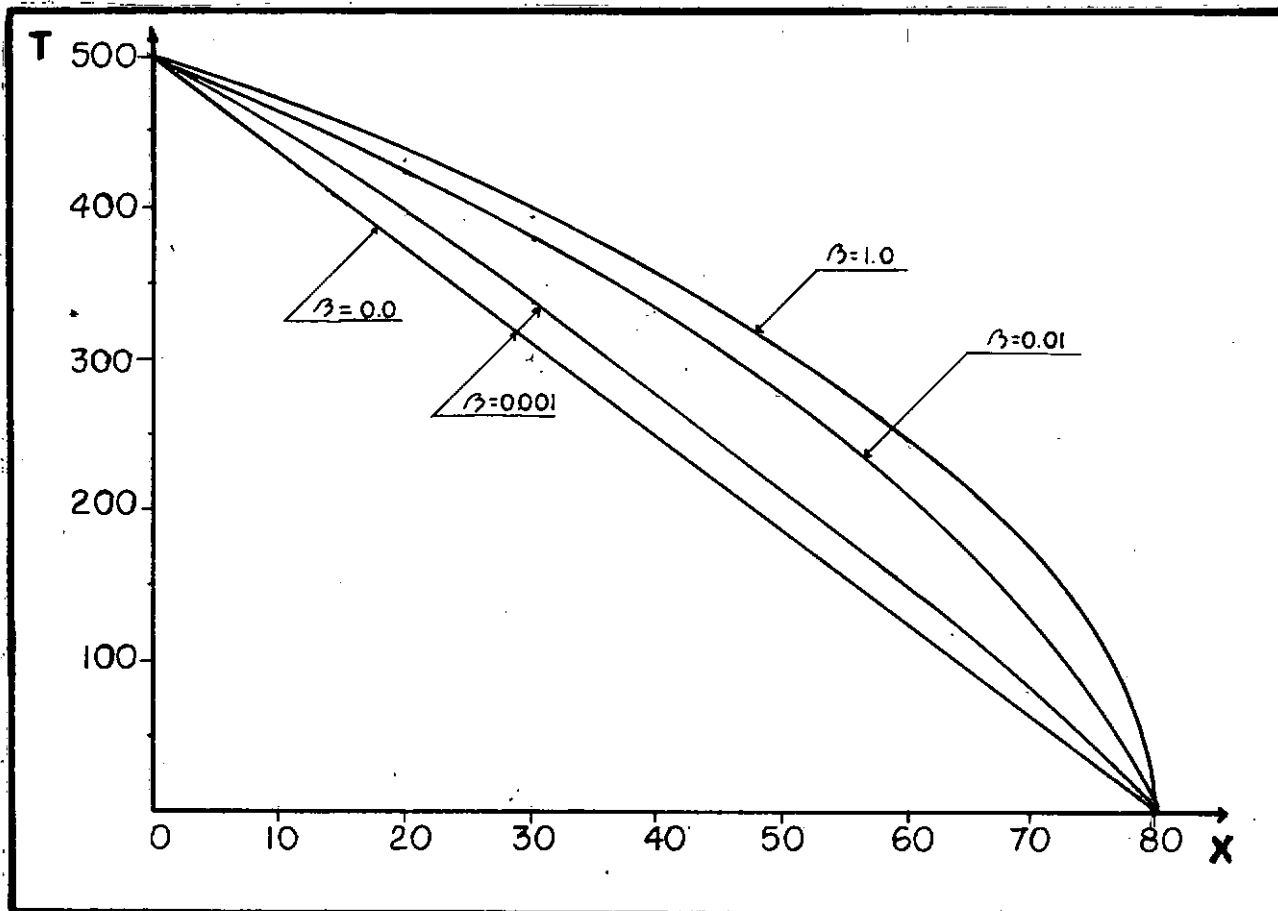


Figura 7.8
Distribuição de Calor para a Placa da Fig. 7.7

7.3.2 - Redes Hidráulicas: Foram incluídos neste trabalho, problemas de análises de redes hidráulicas, que exemplificam de forma clara o uso de métodos lineares. Estes exemplos ainda que não constituam pela formulação em que se baseiam, aplicações do método dos elementos finitos, demonstram a factibilidade do uso de esquemas matriciais que seguem as mesmas linearidades e que estão começando a ser utilizados de forma extensiva pela sua simplicidade e eficiência, neste tipo de problemas.

Para a análise de redes hidráulicas de tubulações foram utilizados elementos lineares, que também se acham implementados no sistema. Esta análise, segundo a formulação em que se baseia, conduz a duas situações diferentes: uma análise linear, apoiada na formulação de Pouseville ou uma análise não-linear com base na formulação de Hazen-Williams [38]. Neste caso, e com a intenção de exemplificar os métodos incrementais e o de Newton-Raphson, utilizou-se o seguinte esquema.

A fórmula de Hazen-Williams é a seguinte:

$$Q = 0.2789 \, c \frac{D^{2,63}}{L^{0,54}} \Delta H^{0,54} \quad (7.6)$$

ou de maneira similar:

$$Q = K \Delta H^{0,54} \quad (7.7)$$

onde Q é o caudal, ΔH a perda de pressão e K o que corresponderia a uma "rigidez", dependente do diâmetro D , o comprimento L e a rugosidade relativa c da tubulação.

Isto conduz a uma formulação matricial dada por elemento na seguinte forma: [38]

$$\begin{Bmatrix} Q_i^k \\ Q_j^k \end{Bmatrix} = K^k \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} H_i \\ H_j \end{Bmatrix} \quad (7.8)$$

onde K^k é um termo que depende da perda de pressão no elemento.

A análise da rede proposta na figura 7.9 se realizou pelo método de Newton-Raphson e pelo método incremental utilizando a matriz tangente. Os resultados se ajustam aos proporcionados pela referência.

As tubulações representadas correspondem a uma rugosidade relativa de 110. Fixaram-se os caudais Q mostrados na figura, e uma altura de pressão $P = 0$ sobre o nó 7.

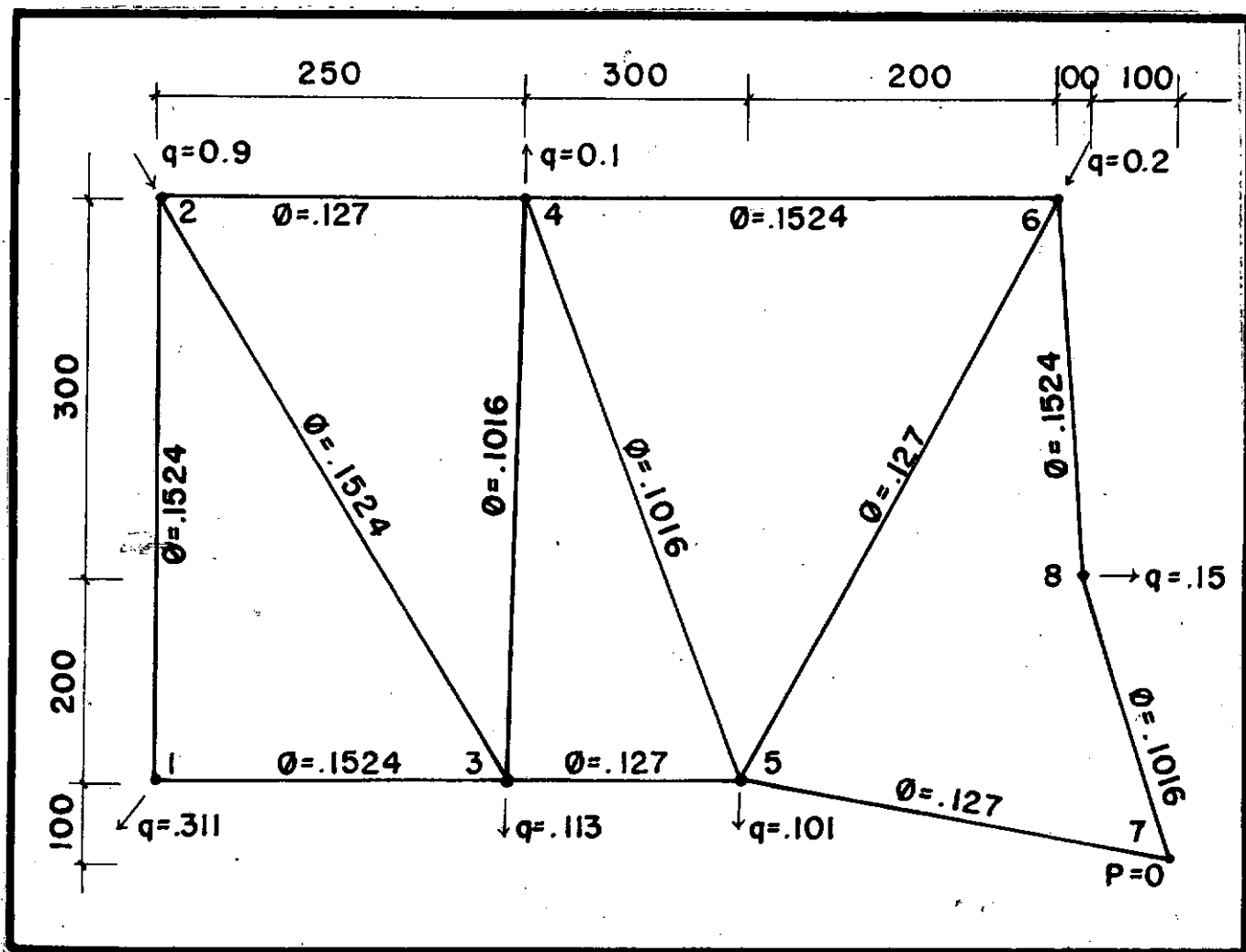


Figura 7.9
Rede de Tubos

10

Nº	ALTURA DA PRESSÃO	Nº	CAUDAL	MEMBRO	CAUDAL
1	1344.1177	1	- .3110	1	- .3145
2	2334.3911	2	.9000	2	.0035
3	1344.0007	3	- .1130	3	.2961
4	1302.4858	4	- .1100	4	.2894
5	896.5677	5	- .1010	5	.0195
6	1088.7588	6	.2000	6	.1670
7	0.0000	7	- .3150	7	.0616
8	647.6539	8	- .1500	8	.1374
				9	- .0772
				10	.2048
				11	- .1102
				12	.2602

Tabela 7.6

Resultados

Os resultados mostrados na Tabela 7.6 correspondem a um Newton Raphson, para uma tolerância de 0.0001. A convergência foi alcançada em 4 iterações.

O mesmo exemplo foi analisado utilizando-se o método incremental, com intervalos distintos de carga, a saber, 10, 20, 30 e 50 intervalos. A Tabela 7.7 mostra as alturas de pressão obtidas para o estado de carga total nos casos anteriores, comparando-os com os de Newton Raphson.

Como se pode ver na Tabela, o erro diminui consideravelmente, de, aproximadamente, 20% para o caso dos 10 intervalos, até 25% para os 50 intervalos.

Nº	NEWTON - RAPHSON	MÉTODO INCREMENTAL							
		10	% Erro	20	% Erro	30	% Erro	50	% Erro
1	1344,1177	1062,9275	20,92	1213,3464	9,73	1266,5071	5,77	1308,7988	2,63
2	2334,3911	1823,5476	22,22	2091,9175	10,77	2187,4805	6,69	2264,1089	3,42
3	1344,0007	1062,6934	23,162	1213,1477	9,74	1266,3235	5,78	1308,6311	2,63
4	1302,4858	1032,8394	20,70	1177,6912	9,58	1228,7891	5,66	1269,3677	2,54
5	896,5677	697,4133	22,21	801,4250	10,61	838,5541	6,47	868,3998	3,14
6	1088,7588	856,9135	21,29	980,0425	9,99	1023,6853	5,98	1058,5154	2,78
7	0,0000	0,0000	0	0,00	0	0,0000	0	0,0000	0
8	647,6539	503,1458	22,31	578,4849	10,68	605,4009	6,52	627,0535	3,18

Tabela 7.7
 Comparações entre o Método Incremental
 e o de Newton - Raphson

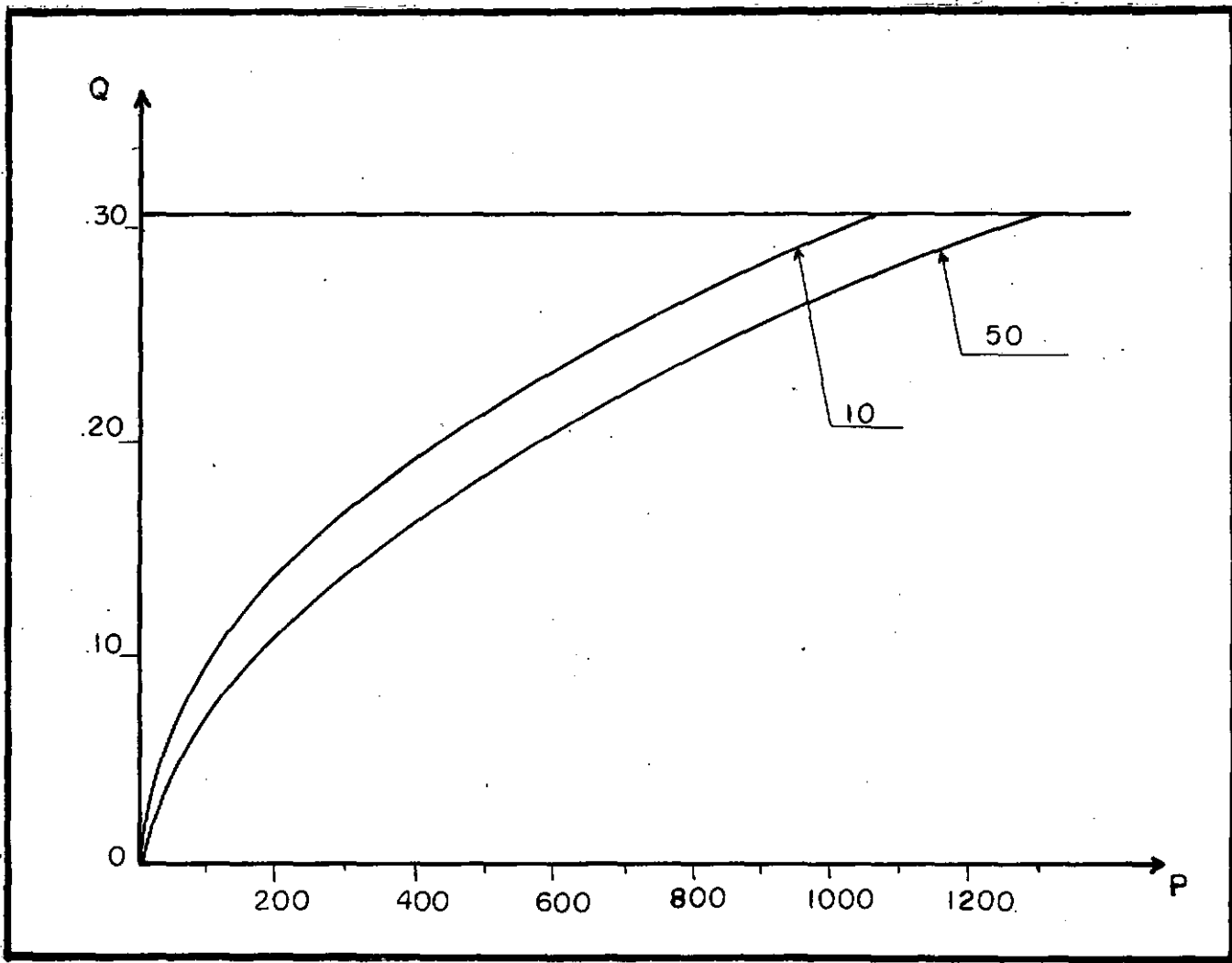


Figura 7.10
Método Incremental

As curvas da Fig. 7.10 mostram a história da carga calculada com base em intervalos ΔQ de $QT/10$ e $QT/50 \text{ m}^3/\text{seg.}$ Devido à forma da função as curvas se aproximam por cima da solução exata. A Tabela 7.8 mostra alguns valores obtidos pelo método incremental para ΔQ de $QT/10$ e $QT/20$.

Como se pode observar nos resultados expostos, o método incremental requer maior esforço computacional para se conseguir resultados de exatidão semelhante ao da teoria de Newton Raphson, ainda que apresente a vantagem de poder se obter através dele toda a história de comportamento de um problema dado.

CAUDAL	PRESSÃO	
	$\Delta Q=Q/10$	$\Delta Q=Q/20$
- 0,0311	13,2141	17,3384
- 0.0622	42,7369	57,0579
- 0.0933	93,5767	121,2788
- 0.1244	166,5646	209,4919
- 0.1555	261,7643	321,1018
- 0.1866	378,9970	455,5693
- 0.2177	518,0028	612,4207
- 0.2488	678,5029	791,2380
- 0.2799	860,2308	991,6554
- 0.3110	1062,9275	1213,3464

Tabela 7.8

Método Incremental

7.4 EXEMPLO DE APLICAÇÃO PARA ANÁLISE POR AUTOVALORES E AUTOVETORES

7.4.1 - Propagação de Ondas em Portos: As fórmulas que regem este problema foram dadas na seção (4.3.4). Para exemplificá-lo, realizou-se a análise do porto de Table Bay, África do Sul, já que foi amplamente estudado [24], [34]. O primeiro período mais relevante é de 11,45 min., segundo as referências.

A malha de elementos finitos utilizada é mostrada na Fig. 7.11 e consta de 120 elementos e 77 nós, para uma largura de banda de 9.

Foram investigadas as 6 primeiras frequências de vibração e autovetores correspondentes, para o que se utilizou o método de iteração por sub-espacos (seção 6.4.2). As iterações se realizaram, todavia, com 12 autovetores, tal como é recomen-

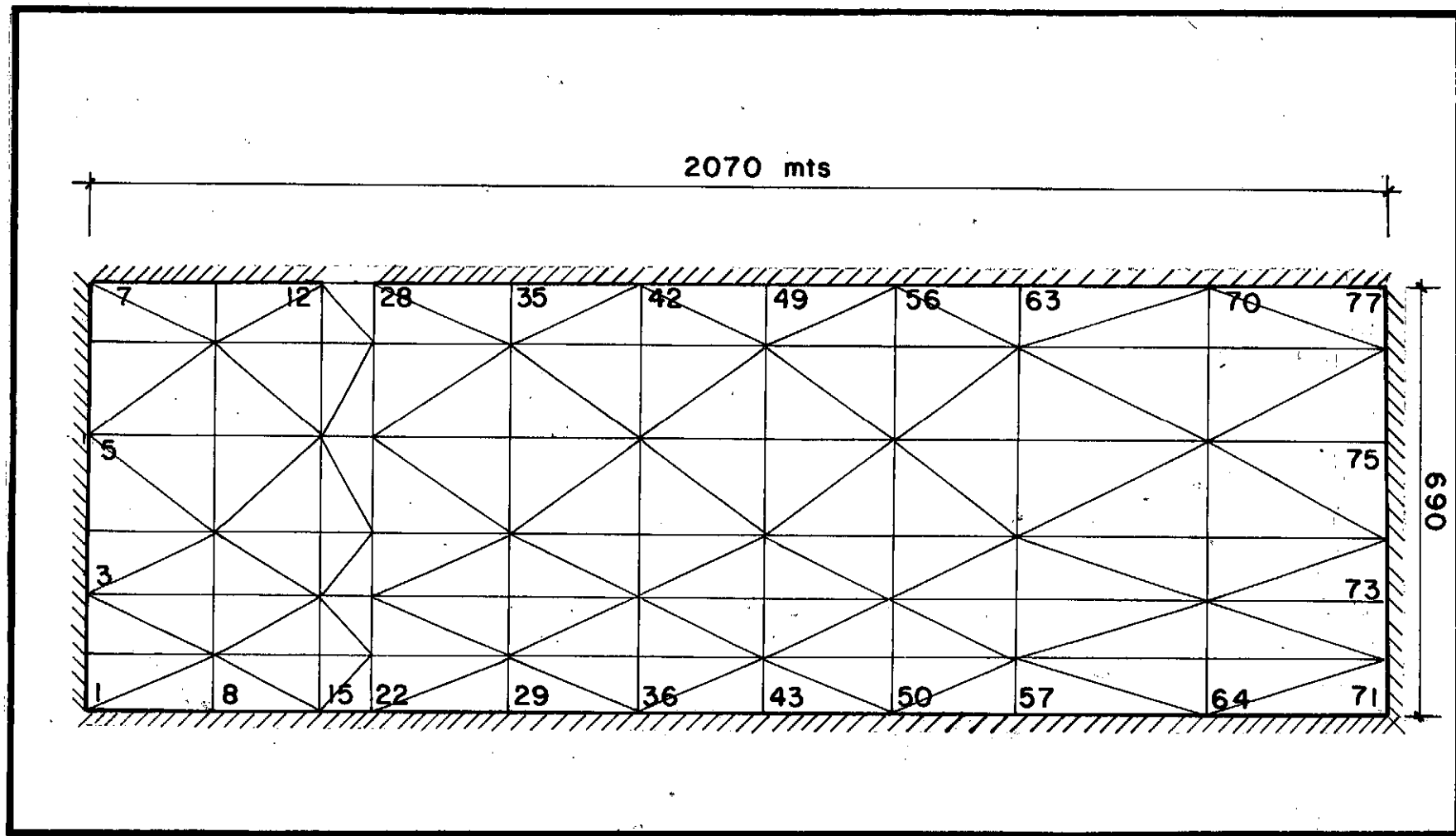


Figura 7.11
Porto de Table Bay

do [26], com o que a convergência foi alcançada em 8 iterações para autovalores com um mínimo de 6 cifras significativas exatas.

O porto, com uma entrada de 85 metros, com um comprimento de 2070 m e uma largura de 690 metros, pode ser aproximado como um corpo fechado de água de baixa profundidade, já que esta só alcança os 12,192 m, aproximadamente.

Para eliminar a singularidade da matriz se fixou a altura da onda sobre o nível do mar na entrada do porto (nós 21 e 28). Isto elimina o modo de vibração correspondente à entrada e saída da água no porto, mas permite conhecer os restantes.

A Tabela 7.9 mostra os autovalores obtidos através do programa, assim como os períodos e as frequências calculadas com base nos mesmos.

AUTOVALORES	T(min)	ω (rad/min)
$0,7664 \ 5556 \cdot 10^{-5}$	11,96	0,0836
$0,5141 \ 9942 \cdot 10^{-4}$	4,6181	0,2165
$0,1198 \ 4029 \cdot 10^{-3}$	3,0250	0,3306
$0,2669 \ 302 \cdot 10^{-3}$	2,0269	0,4934
$0,2786 \ 5247 \cdot 10^{-3}$	1,9838	0,5041
$0,3546 \ 6888 \cdot 10^{-3}$	1,7584	0,5687

Tabela 7.9
Resultados: Porto de Table Bay

O erro cometido para a malha empregada, comparando o resultado dado na Tabela 7.7 para o primeiro período com o valor dado como correto de 11,96 min, é de 45%, o que se considera bastante aceitável, sobretudo se se leva em conta as simplificações adotadas na representação da forma e profundidade da baía.

NºS	T = 11,96	T = 4,62	T = 3,03	T = 2,03	T = 1,98	T = 1,76
1	$0.4054 \cdot 10^{-3}$	$-0.1633 \cdot 10^{-2}$	$-0.155 \cdot 10^{-2}$	$+0.3214 \cdot 10^{-3}$	$-0.7658 \cdot 10^{-3}$	$-0.1399 \cdot 10^{-2}$
8	$0.4185 \cdot 10^{-3}$	$-0.1566 \cdot 10^{-2}$	$-0.131 \cdot 10^{-2}$	$+0.8146 \cdot 10^{-4}$	$-0.3628 \cdot 10^{-3}$	$-0.1373 \cdot 10^{-2}$
15	$0.4635 \cdot 10^{-3}$	$-0.1422 \cdot 10^{-2}$	$-0.7378 \cdot 10^{-3}$	$+0.4745 \cdot 10^{-3}$	$+0.5069 \cdot 10^{-3}$	$-0.1414 \cdot 10^{-2}$
22	$0.4956 \cdot 10^{-3}$	$-0.1334 \cdot 10^{-2}$	$-0.4075 \cdot 10^{-3}$	$-0.7339 \cdot 10^{-3}$	$+0.868 \cdot 10^{-3}$	$-0.1336 \cdot 10^{-2}$
29	$0.6181 \cdot 10^{-3}$	$-0.1065 \cdot 10^{-2}$	$+0.4265 \cdot 10^{-3}$	$-0.115 \cdot 10^{-2}$	$+0.1215 \cdot 10^{-2}$	$-0.1025 \cdot 10^{-2}$
36	$0.7452 \cdot 10^{-3}$	$-0.7705 \cdot 10^{-3}$	$+0.9957 \cdot 10^{-3}$	$-0.118 \cdot 10^{-2}$	$+0.5542 \cdot 10^{-3}$	$-0.9412 \cdot 10^{-3}$
43	$0.8707 \cdot 10^{-3}$	$-0.4063 \cdot 10^{-3}$	$+0.1108 \cdot 10^{-2}$	$-0.942 \cdot 10^{-3}$	$-0.6786 \cdot 10^{-3}$	$-0.9253 \cdot 10^{-3}$
50	$0.9844 \cdot 10^{-3}$	$-0.1230 \cdot 10^{-4}$	$+0.8106 \cdot 10^{-3}$	$-0.8604 \cdot 10^{-3}$	$-0.1675 \cdot 10^{-2}$	$-0.8277 \cdot 10^{-3}$
57	$0.1076 \cdot 10^{-2}$	$+0.3677 \cdot 10^{-3}$	$+0.1889 \cdot 10^{-3}$	$-0.1033 \cdot 10^{-2}$	$-0.1706 \cdot 10^{-2}$	$-0.2159 \cdot 10^{-3}$
64	$0.1168 \cdot 10^{-2}$	$+0.8002 \cdot 10^{-3}$	$+0.8061 \cdot 10^{-3}$	$-0.1771 \cdot 10^{-2}$	$-0.5065 \cdot 10^{-3}$	$+0.1194 \cdot 10^{-2}$
71	$0.1194 \cdot 10^{-2}$	$+0.9346 \cdot 10^{-3}$	$+0.1172 \cdot 10^{-2}$	$-0.2056 \cdot 10^{-2}$	$-0.2176 \cdot 10^{-3}$	$+0.1917 \cdot 10^{-2}$

Tabela 7.10
Elevações de Crista em Metros

NÓS	T = 11,96	T = 4,6168	T = 3,0250	T = 2,0269	T = 1,9838	T = 1,7584
71	$0.1194 \cdot 10^{-2}$	$0.9346 \cdot 10^{-3}$	$-0.1172 \cdot 10^{-3}$	$-0.2056 \cdot 10^{-2}$	$0.2176 \cdot 10^{-3}$	$-0.1817 \cdot 10^{-2}$
72	$0.1195 \cdot 10^{-2}$	$0.9403 \cdot 10^{-3}$	$-0.1187 \cdot 10^{-2}$	$-0.1952 \cdot 10^{-2}$	$0.3123 \cdot 10^{-3}$	$0.1744 \cdot 10^{-2}$
73	$0.1193 \cdot 10^{-2}$	$0.9359 \cdot 10^{-3}$	$-0.1172 \cdot 10^{-2}$	$-0.1444 \cdot 10^{-2}$	$0.5783 \cdot 10^{-3}$	$0.1302 \cdot 10^{-2}$
74	$0.1194 \cdot 10^{-2}$	$0.9429 \cdot 10^{-3}$	$-0.1187 \cdot 10^{-2}$	$-0.8176 \cdot 10^{-3}$	$0.9222 \cdot 10^{-3}$	$0.7824 \cdot 10^{-3}$
75	$0.1192 \cdot 10^{-2}$	$0.9343 \cdot 10^{-3}$	$-0.1158 \cdot 10^{-2}$	$+0.2512 \cdot 10^{-3}$	$0.1534 \cdot 10^{-2}$	$-0.1186 \cdot 10^{-3}$
76	$0.1194 \cdot 10^{-2}$	$0.9458 \cdot 10^{-3}$	$-0.1185 \cdot 10^{-2}$	$+0.1026 \cdot 10^{-2}$	$0.2064 \cdot 10^{-2}$	$-0.7721 \cdot 10^{-3}$
77	$0.1193 \cdot 10^{-2}$	$0.9411 \cdot 10^{-3}$	$-0.1172 \cdot 10^{-2}$	$+0.1126 \cdot 10^{-2}$	$0.2092 \cdot 10^{-2}$	$-0.8522 \cdot 10^{-3}$

Tabela 7.11
Elevações de Crista em Metros
sobre o Bordo Direito

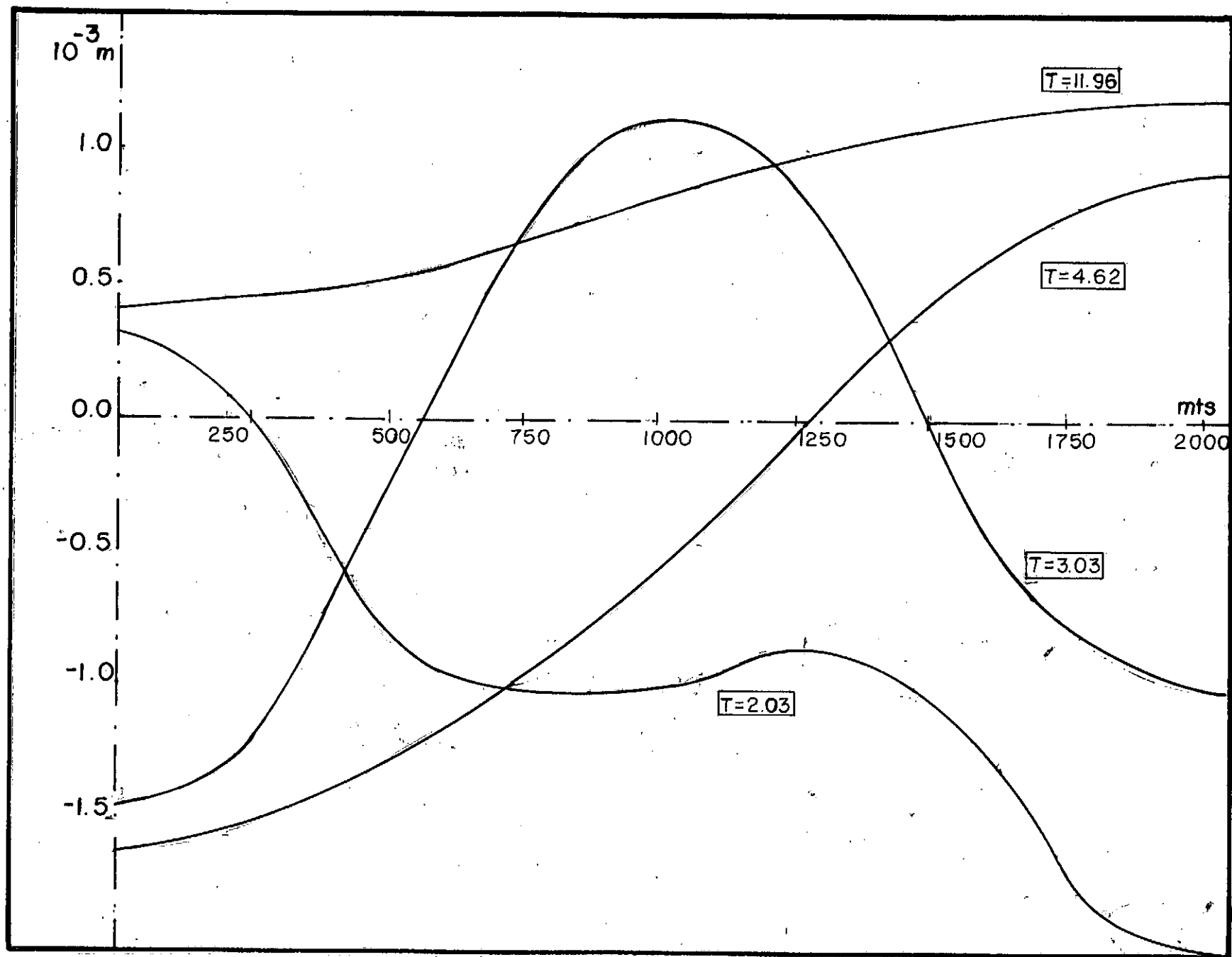


Figura 7.12

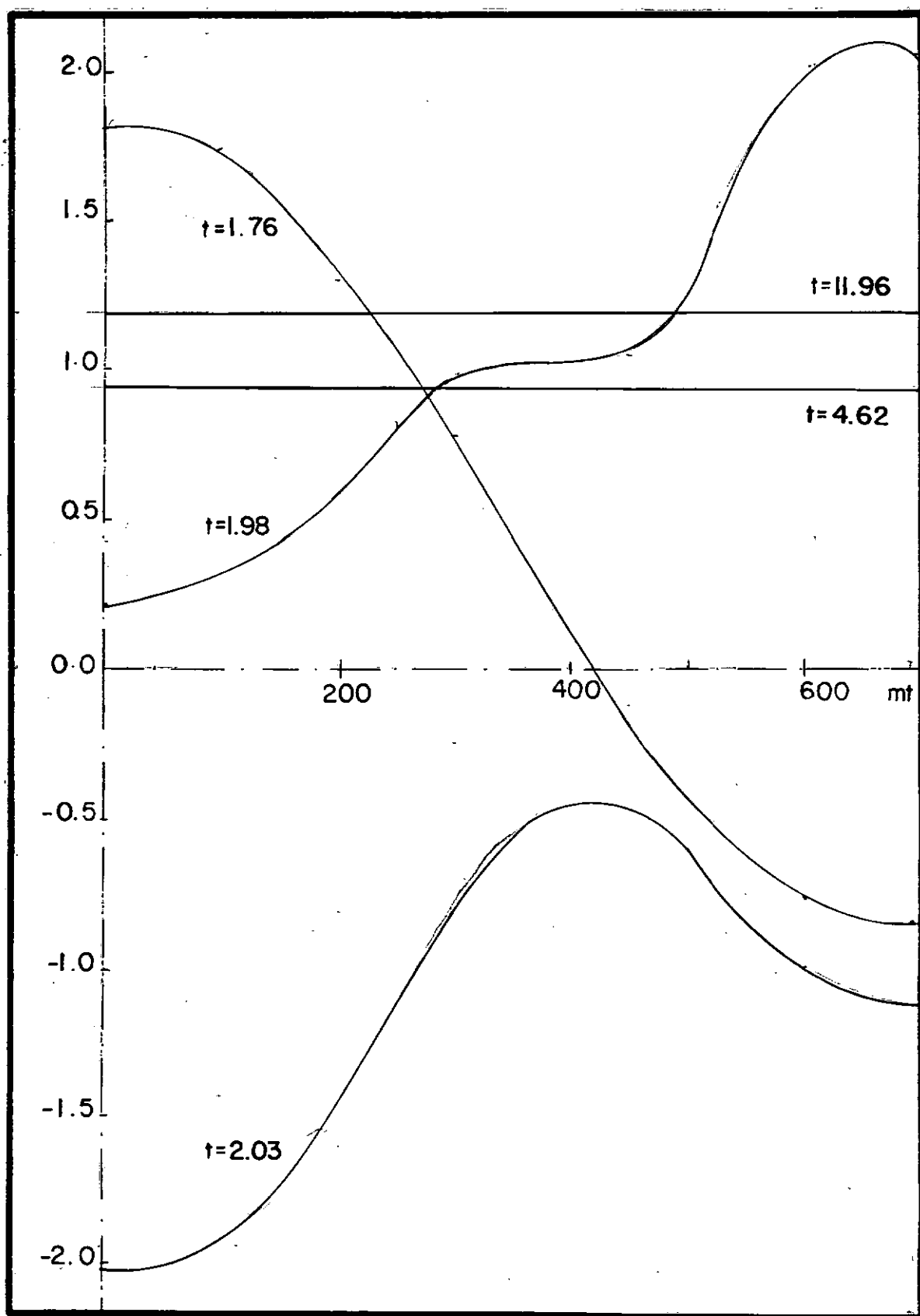


Figura 7.13
Elevação da Crista sobre o Bordo

O conhecimento destes valores de frequências correspondentes às configurações de menor energia são convenientes na hora de projetar o porto, já que se pode evitar com isso, problemas de ressonância com a onda incidente.

A Tabela 7.10 mostra os valores da elevação da crista da onda sobre o nível do mar para os nós colocados no bordo oposto à entrada do porto, e para as frequências calculadas.

Estes valores são postos em gráfico na Fig. 7.12. Construi-se uma Tabela semelhante para os nós sobre o bordo direito do porto (Tabela 7.11) e se fizeram gráficos na Figura 7.13.

7.4.2 - Propagação de Ondas Acústicas: As fórmulas que regem este problema foram dadas na seção (4.3.4).

Neste caso se tratou de representar a cabine de um automóvel, para estudar como se propagam as ondas sonoras em seu interior. Da mesma maneira que no caso anterior, empregou-se o método de iteração por sub-espacos com 12 vetores de iteração para se obter os seis primeiros autovalores e autovetores. Para uma precisão no sexto autovalor de 0.000001 foram empregadas 9 iterações.

A malha utilizada consta de 53 nós para um total de 78 elementos triangulares, tal como se mostra na Fig. 7.14. Os autovalores obtidos se encontram na Tabela 7.12 e os autovetores na Tabela 7.13.

CABINA DE AUTOMOVIL

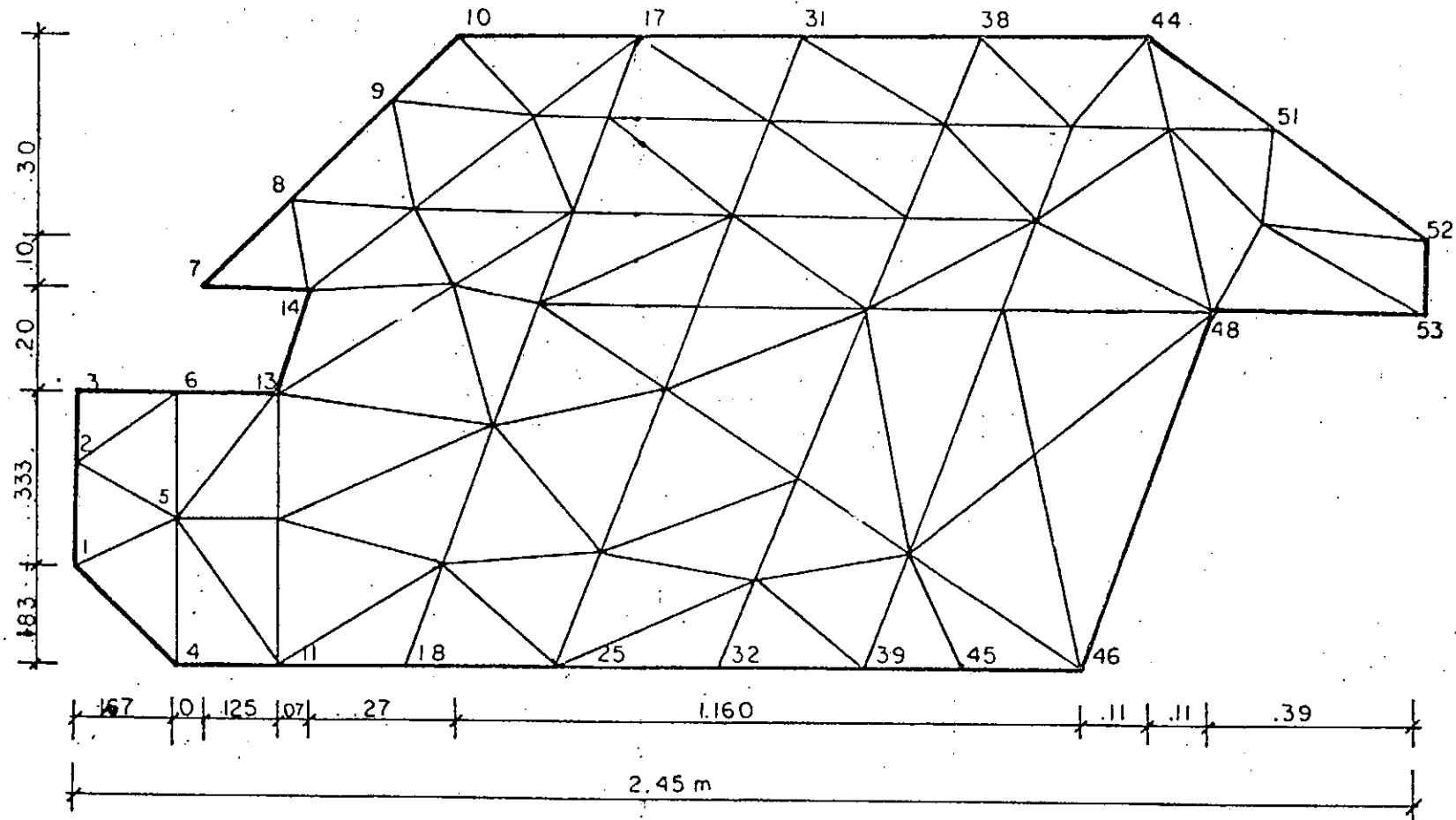


Figura 7.14
Discretização do Modelo

MODO DE VIBRAÇÃO	(rad/seg)
1	217,90
2	581,97
3	912,40
4	1107,43
5	1347,07
6	1635,26

Tabela 7.12

Autovalores para a Cabine
de Automóvel

7.5 EXEMPLO DE ENTRADA DE DADOS UTILIZANDO A LINGUAGEM ORIENTADA

Em seguida são incluídos exemplos de entradas de dados que ilustram a utilização dos diferentes comandos da linguagem orientada. O segundo exemplo corresponde à entrada de dados para o problema de propagação de ondas acústicas no interior da cabine de um automóvel, problema cujos resultados foram apresentados na seção anterior. O primeiro exemplo corresponde ao problema de distribuição de calor em uma placa quadrada (seção 7.3.1) e ilustra a utilização do comando superelemento numa geração da malha da Figura 7.6.

```

TITULO
PLACA CUADRADA SOMETIDA A AMBIOS DE TEMPERATURA
RESTRICCIONES NODALES
  1 6 11 16 21 F 500. 10 15 20 25 F 0
TIPO LAPLACE
ANALISIS NO LINEAL
CONSTANTES J KX 1. KY 1. C 0. LAMBDA 0
SUPERELEMENTO
  1 ELEMENTOS LINEALES E 4 POR 4
  0. 0. 80. 0. 80. 80. 0. 80.
BETA 1.
FIN

```

Figura 7.15

Entrada de Dados Nº 1

LOS AUTOVALORES CALCULADOS SON:

410733760000000E+00 .292979760000000E+01 .720138650000000E+01
 106090130000000E+02 .156972080000000E+02 .231321140000000E+02

LOS AUTOVECTORES CALCULADOS SON:

.8703E+00	-.1131E+01	-.1467E+01	-.3563E+00	.1055E+01	.9742E+00
.8723E+00	-.1140E+01	-.1479E+01	-.4807E+00	.1117E+01	.1496E+01
.8714E+00	-.1130E+01	-.1446E+01	-.4951E+00	.1083E+01	.1508E+01
.8639E+00	-.1087E+01	-.1352E+01	-.1549E+00	.8313E+00	.1772E+00
.8664E+00	-.1092E+01	-.1339E+01	-.3436E+00	.8714E+00	.8226E+00
.8615E+00	-.1028E+01	-.1118E+01	-.4797E+00	.6367E+00	.1068E+01
.8234E+00	-.5187E+00	.7373E+00	-.1372E+01	-.1738E+01	.1358E+01
.8170E+00	-.4806E+00	.6979E+00	-.1187E+01	-.1364E+01	.9539E+00
.8029E+00	-.3268E+00	.9649E+00	-.1277E+01	-.1090E+01	.8317E+00
.7850E+00	-.1517E+00	.1157E+01	-.1280E+01	-.5832E+00	.5795E+00
.8484E+00	-.9661E+00	-.1018E+01	.1253E+00	.3184E+00	-.8313E+00
.8463E+00	-.9195E+00	-.8378E+00	-.1065E+00	.1698E+00	-.2443E+00
.8395E+00	-.8037E+00	-.4180E+00	-.4681E+00	-.2376E+00	.3046E+00
.8168E+00	-.4967E+00	.6026E+00	-.1068E+01	-.1222E+01	.7970E+00
.8107E+00	-.4289E+00	.7497E+00	-.1114E+01	-.1187E+01	.7483E+00
.7842E+00	-.1692E+00	.1086E+01	-.1166E+01	-.5936E+00	.5101E+00
.7486E+00	.1339E+00	.1114E+01	-.9611E+00	.5105E+00	-.1086E+00
.8250E+00	-.7989E+00	-.6088E+00	.4550E+00	-.1743E+00	-.1492E+01
.8152E+00	-.7244E+00	-.4245E+00	.5296E+00	-.3574E+00	-.1574E+01
.7957E+00	-.5234E+00	.1078E+00	.2321E+00	-.6505E+00	-.9396E+00
.8064E+00	-.4714E+00	.4728E+00	-.6414E+00	-.9060E+00	.2002E+00
.7657E+00	-.1036E+00	.9151E+00	-.6620E+00	-.2413E+00	.1811E-01
.7575E+00	-.2793E-01	.1066E+01	-.8934E+00	.1428E+00	.3807E-01
.7757E+00	-.2942E+00	.6002E+00	-.1894E+00	-.5805E+00	-.2699E+00
.7712E+00	-.4914E+00	-.4050E-01	.1148E+01	-.6370E+00	-.1726E+01
.7400E+00	-.2989E+00	.2576E+00	.1061E+01	-.5802E+00	-.1014E+01
.7147E+00	-.5783E-01	.5465E+00	.6211E+00	-.1826E+00	-.3093E+00
.7122E+00	.6689E-01	.6641E+00	.2689E+00	.1730E+00	-.1584E+00
.7042E+00	.2754E+00	.8067E+00	-.1884E+00	.7577E+00	-.1767E+00
.7000E+00	.4202E+00	.7702E+00	-.4427E+00	.1209E+01	-.4857E+00
.6874E+00	.5640E+00	.6189E+00	-.4749E+00	.1507E+01	-.7721E+00
.6956E+00	-.1885E+00	.2587E+00	.1521E+01	-.5237E+00	-.4840E+00
.6637E+00	-.5014E-01	.3816E+00	.1513E+01	-.3969E+00	.8736E-01
.6384E+00	.1445E+00	.4696E+00	.1098E+01	.3274E-01	.5190E+00
.6308E+00	.4148E+00	.4408E+00	.4967E+00	.7018E+00	.4547E+00
.6421E+00	.6332E+00	.2992E+00	.1659E-01	.1079E+01	-.1735E+00
.6414E+00	.8171E+00	.1002E+00	-.2370E+00	.1312E+01	-.6963E+00
.6338E+00	.9583E+00	-.1411E+00	-.3172E+00	.1233E+01	-.9495E+00
.5408E+00	.1217E+00	.3309E+00	.1610E+01	-.2376E+00	.1324E+01
.4976E+00	.3074E+00	.2427E+00	.1198E+01	.5432E-01	.1509E+01
.5406E+00	.7253E+00	-.1092E+00	.2673E+00	.4601E+00	.5532E+00
.5847E+00	.9467E+00	-.3115E+00	-.6837E-01	.6013E+00	-.7855E-01
.6095E+00	.1069E+01	-.4364E+00	-.2721E+00	.5997E+00	-.6231E+00
.6110E+00	.1225E+01	-.7706E+00	-.3989E+00	.1397E+00	-.8283E+00
.3803E+00	.1882E+00	.2153E+00	.1092E+01	-.3442E-01	.1392E+01
.4093E+00	.5416E+00	-.1344E+00	.4233E+00	.4814E-01	.8827E+00
.5674E+00	.1178E+01	-.9038E+00	-.1613E+00	-.8178E+00	.3797E+00
.6025E+00	.1255E+01	-.8982E+00	-.3795E+00	-.3699E+00	-.4731E+00
.5906E+00	.1341E+01	-.1189E+01	-.4052E+00	-.1397E+01	-.4769E-03
.5964E+00	.1331E+01	-.1126E+01	-.4207E+00	-.1099E+01	-.2135E+00
.5969E+00	.1453E+01	-.1455E+01	-.5415E+00	-.2216E+01	.4674E-01
.5987E+00	.1480E+01	-.1516E+01	-.5735E+00	-.2397E+01	.5802E-01
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00

TABELA 7.13

Autovetores Da Cabina Do Automovil

TITULO

* CABINA DE AUTOMOVIL *

TIPO HEMHOLTZ

ANALISIS VIBRACIONES LIBRES AUTOVALORES Y AUTOVECTORES

CONSTANTES

KY 1. KX 1. C 0.

CONECTIVIDADES

1 1 4 5 / 2 1 5 3 / 3 2 5 6 / 4 2 6 3 / 5 4 11 5 / 6 11 12 5 / 7C
 5 12 3 / 8 5 13 6 / 9 11 19 12 / 10 11 18 19 / 11 12 19 20
 12 12 20 13 / 13 13 20 21 / 14 13 21 14 / 15 14 21 15
 COORDENADAS / 1 X 0. Y .2 / 2 0. .362 / 3 X 0. Y 5.22E-1
 4 .16 0. / 5 .180 .3 / 6 .2 Y .54 / 7 X .22 Y .63
 8 Y .84 X .36 / 9 Y .98 X .52 / 10 7E-1 .114E1
 11 .38 Y 0.
 12 X .38 Y .3 / 13 X .38 Y .54 / 14 .44 .74 / 15 5.2E-1 Y .8
 16 .74 .96

RESTRICCIONES NODALES

46 F 0.

CONECTIVIDADES / 16 7 14 8 / 17 14 15 8 / 18 8 15 9
 19 15 16 9 / 20 9 16 10 / 21 16 17 10 / 22 16 23 17
 23 16 22 23 / 24 15 22 16 / COORDENADAS / 17 1.01 1.14
 COORDENADAS / 18 .54 0. / 19 .6 .14 / 20 .7 .4
 21 X .6 Y .65 / 22 .88 .8 / 23 Y .96 X .96
 24 Y .6 X .8 / 25 .0 0. / 26 .92 .26
 27 1.04 .48 / 28 10.86E-1 .610
 29 1.18 .8 / 30 1.25 .96 / 31 1.342 1.14
 32 1.04 0. / 33 1.14 .19 / 34 1.244 .39 / 35 1.364 .610
 36 1.46 .8 / COORDENADAS / 37 1.56 .96 / 38 X 1.66 1.14
 39 1.38 / 40 1.5 .266 / 41E1 1.66 C
 42 1.74 .79 / 43 1.8 .95 / 44 1.98 1.14 / 45 1.59 C
 .04 / 46 1.86 Y .102 / 47 1.73 .41 / 48 2.08 .62
 49 2.02 .942 / 50 2.21 .79 / 51 X 2.21 Y .94
 52 2.45 .747 / 53 2.45 .62
 CONECTIVIDADES / 16 7 14 8 / 17 14 15 8 / 18 8 15 9
 19 15 16 9 / 20 9 16 10 / 21 16 17 10 / 22 16 23 17
 23 16 22 23 / 24 15 22 16 / 25 21 22 15 / 26 21 24 22
 27 20 24 21 / 28 18 25 19 / 29 19 25 26 / 30 19 26 20
 31 20 26 27 / 32 20 27 24 / 33 24 27 28 / 34 24 28 29
 35 24 29 22 / 36 22 29 23 / 37 23 29 30 / 38 23 30 17
 39 30 31 17 / 40 25 32 33 / 41 25 33 26 / 42 26 33 34
 43 26 34 27 / 44 27 34 28 / 45 28 34 35 / 46 28 35 29
 47 29 35 36 / 48 29 36 30 / 49 30 36 37 / 50 30 37 31
 51 37 38 31 / 52 32 39 33 / 53 39 40 33 / 54 33 40 34
 55 34 40 35 / 56 40 41 35 / 57 35 41 42 / 58 35 42 36
 59 36 42 37 / 60 42 43 37 / 61 37 43 38 / 62 43 44 38
 63 39 45 40 / 64 45 46 40 / 65 40 46 47 / 66 40 47 41
 67 47 46 48
 68 47 48 41 / 69 41 48 42 / 70 42 48 49 / 71 42 49 43
 72 43 49 44 / 73 49 51 44 / 74 50 51 49 / 75 48 50 49
 76 48 53 50 / 77 50 53 52 / 78 50 52 51
 IMPRESION DE DATOS
 IMPRESION DE RESULTADOS INTERMEDIOS Y FINALES
 FIN

Figura 7.17

Entrada de Datos n° 2

CONCLUSÕES E COMENTÁRIOS

1 - O desenvolvimento e o crescente uso que vem sendo dado aos computadores em diferentes áreas, tanto de pesquisa como educacionais, e ainda na prática diária do engenheiro moderno, leva à necessidade de se adaptar grandes programas, que tradicionalmente só eram criados para computadores maiores, para estas máquinas. Nos últimos tempos têm aparecido diversas linguagens e sistemas baseados em elementos finitos, adaptados à minicomputadores e com capacidade de resolver grandes problemas. Na Venezuela esta área, como outras nos diferentes ramos de avanço da tecnologia, vem sendo desprezada, dando-se pouca ou nenhuma importância à criação de pacotes próprios adaptados às necessidades específicas de companhias, institutos de pesquisas e Universidades venezuelanas. Neste sentido, este trabalho constitui um pequeno passo na divulgação da existência, capacidade e utilidade de um sistema para minicomputadores baseado no método dos elementos finitos, e da forma com que se estrutura e organiza como tal.

2 - O sistema provou ser bastante eficiente no manejo da informação armazenada em disco, solucionando completamente o problema da interfase para o usuário e permitindo a resolução de problemas de grande dimensão. As estratégias adotadas na recuperação de informações durante a solução do sistema de equações através da função Sapo diminuem consideravelmente o tempo de execução. Ao mesmo tempo a utilização de um sistema que simula memória virtual permite uma grande versatilidade para o armazenamento de dados. Este duplo aspecto da comunicação com o disco rendeu resultados satisfatórios.

3 - A linguagem orientada constitui um sistema de comunicação com o usuário que permite um uso eficaz e completo das potencialidades do mesmo, além de facilitar sua expansão através da criação de novos comandos. O uso de uma linguagem orientada simplifica a entrada de dados para o usuário comum e facilita a aprendizagem de sua utilização. Estas características

diversificam e amplificam o seu uso, expandindo-o, por exemplo, dentro de esferas educacionais.

4 - O sistema apresentado neste trabalho possui as importantes características de modularidade e compatibilidade. É compatível no sentido de que está elaborado num FORTRAN básico, o que permite transferir o pacote para diversos computadores, alterando somente pequenos detalhes. Estes detalhes, além disso, (como por exemplo as instruções sobre o manejo do disco, que variam em cada máquina), encontram-se localizados em setores específicos do programas, para facilitar sua colocação. O sistema por outro lado, está estruturado com bases em módulos que permitem sua expansão progressiva, sem alterar o funcionamento de módulos já existentes.

5 - Os módulos de aplicação disponíveis permitem a realização de análises linear e não linear e de autovalores e autovetores. Suas características de generalidade permitem utilizá-los na solução de problemas de diversas áreas, assim como os baseados na equação de campo, como a maioria dos apresentados neste trabalho, redes hidráulicas (ver exemplo de aplicação) e diversos problemas de estruturas.

6 - A estrutura do sistema torna-se suficientemente flexível para permitir a adição de qualquer novo problema de aplicação ainda não considerado. Esta nova aplicação pode ser implementada com base em algumas rotinas já existentes de solução ou análises baseadas nos sistemas de comunicação com o disco, já que estas rotinas e sistemas são completamente gerais e se utilizam apoiadas em alguns parâmetros fundamentais. A utilização do novo módulo requererá também a adição do comando correspondente à linguagem orientada.

7 - Neste trabalho as aplicações realizadas se referem em sua maioria a problemas derivados da equação de campo generalizada. A resolução deste tipo de problemas pelo método dos elementos finitos abre a possibilidade de exploração de novas áreas não tratadas tradicionalmente por este tipo de método: hidráulica, eletro-magnetismo, transferencia de calor, etc.

8 - As limitações do sistema, quanto ao número de métodos de análises de que dispõem e quanto à potencialidade de sua linguagem orientada, não são tantas, que no que se refere à sua estrutura e versatilidade não possam ser facilmente ampliadas, em fases posteriores. Em uma próxima etapa, por exemplo, encontra-se a implementação de módulos que permitem a realização de análises de problemas dependentes do tempo, módulos de geração de malhas mais sofisticados, módulos de post-processamento, incluindo rotinas de projeto, etc.

9 - Por último, é válido recordar que a utilização de qualquer programa não deve ser feita às cegas, sem que se conheçam detalhes de sua implementação, características, limitações, etc., porque pode levar a surpresas desagradáveis. Da mesma forma, deve se conhecer e dominar o problema físico que se está resolvendo, para poder aproveitar ao máximo as potencialidades disponíveis do sistema.

10 - Nos Estados Unidos e Europa se encontra muito difundido o uso de minicomputadores em diversas áreas da engenharia. Se bem que na Venezuela, e em outros países da América do Sul este uso tenha sido muito limitado, a tendência é de que se incremente cada vez mais, pelas exigências que o desenvolvimento destes países impõe.

No entanto, o fato de se tratar de minicomputadores impõe certas limitações, e existe uma dúvida razoável sobre se se poderia usá-los ou não, para resolver problemas relevantes na prática da engenharia. Estas limitações se referem à dimensão da palavra, à capacidade da memória e a velocidade do cálculo.

A limitação da longitude da palavra traz inconvenientes, sobretudo a nível dos inteiros, já que quase não se permitem de 4 bytes, mas isto tende a ser superado com a aparição de compiladores mais eficientes, por exemplo FORTRAN IV PLUS e BASIC II PLUS, de PDP. Quanto a números reais, isto não parece ser inconveniente maior, já que geralmente se permitem variáveis de 4 e 8 bytes por software, ou senão, opcionalmente, por

hardware. O maior problema então, da palavra de 16 bits: é a consequente limitação no direcionamento, mas também isto tende a ser superado com técnicas tipo "overlay" ou outras.

Quanto à capacidade de memória, deve notar-se que ultimamente foram introduzidos minicomputadores de baixo e médio preço com capacidade de memória direta de 1 μ byte ou mais, por exemplo, PDP 11/24 e PDP 11/44. Ainda que com memórias menores, esquemas do tipo arranjo virtual permitem utilizar em FORTRAN, arranjos com dimensões da ordem de 32000 palavras, não muito diferentes às limitações em computadores grandes, como o BURROUGHS B6700. Por outro lado, estes problemas se resolvem eficientemente com rotinas adequadas para o manejo de informações em memória secundária, sendo justamente, um dos objetivos desta tese, o estudo de técnicas deste tipo.

Quanto à velocidade de cálculo, deve se fazer notar que os processadores dos minicomputadores são bastante velozes. Por exemplo, a velocidade de processamento de uma PDP 11/44 é igual ou maior que a de um computador IBM 370 de tamanho médio, ou da BURROUGHS B6700. Também deve-se considerar que não só a velocidade de cálculo, como também o tempo de resposta é um fator importante para problemas de engenharia. No caso de um minicomputador usado através de terminais, esquema para o qual foi especialmente projetado, o tempo de resposta é geralmente melhor que em um sistema Batch com computador grande.

No presente trabalho foi demonstrada a viabilidade de implementar um sistema de computador versátil e eficaz na resolução de variados problemas de engenharia, em um minicomputador com as limitações mencionadas, obtendo-se resultados satisfatórios. De fato, foi possível que no minicomputador a utilização de uma linguagem orientada o suficientemente potente para resolver problemas muito variados de engenharia, de relevância prática. Os problemas enfrentados alcançam dimensões respeitáveis, chegando-se a resolver matrizes de 34000 elementos em tempos razoáveis para um computador pequeno.

BIBLIOGRAFIA

1. GRANINO A. KORN, "Minicomputers for Engineers and Scientists", McGraw-Hill Book Company, 1973.
2. RICHARD H. ECKHOUSE Jr, "The Small Computer System: Organization and Programming", Dezembro, 1973.
3. HEWLETT-PACKARD: "Computer Programming Course", 1973.
4. HEWLETT-PACKARD 2100s "Microprogrammable Systems Computer", Reference Manual, Abril, 1973.
5. PDP, "Small Computer Handbook", Digital, 1973.
6. PDP 11, "Processor Handbook", Digital, 1976.
7. HEWLETT-PACKARD 1000 Computers and Systems, "ACTIVE SOFTWARE DATA", 1979.
8. HEWLETT-PACHARD programming and operating manual, "Real time Executive II Software System", 1975.
9. "HP 1000 M/E/F - Series Computers Technical Reference Handbook", 1979
10. HP "ELECTRONIC INSTRUMENTS AND SYSTEMS", 1981.
11. M. LAFUENTE, M. CERROLAZA, A.J.FERRANTE, "Sistemas de Mini-computador para Engenharia Civil". Anais do I Colóquio Brasil Venezuela de Engenharia Civil, 1980.
12. N. CAMPOS, E. CASTILLA, A. GUELL, "Manual del Programador de SMD-2". Instituto de Materiales y Modelos Estructurales, UCV; 1979.

13. N. CAMPOS, E. CASTILLA, A. GUELL, "Manual del Usuario de SMD-2", Instituto de Materiales y Modelos Estructurales, UCV, 1979.
14. E. CASTILLA, A. GUELL, "Software pra Memoria Virtual em FORTRAN", III Simpósio sobre Sistemas Computacionais para Engenharia Civil - Anais - 1979.
15. RICHARD H. ECKHOUSE Jr., L.R. MORRIS, "Minicomputer Systems: Organization, Programming and Applications (PDP - 11)", Prentice -Hall, 1979.
16. D.D. SPENCER "Computers in Action", Hayden Book Company, Inc., 1978.
17. M. LAFUENTE, M. CERROLAZA, A.J. FERRANTE, "Desenvolvimento de Programas de Elementos Finitos para Minicomputador", III Simpósio sobre Sistemas Computacionais para Engenharia Civil, Anais, 1979.
18. O.C. ZIENKIEWICZ, "The Finite Element Method", McGraw-Hill Book Company, 1977.
19. O.C. ZIENKIEWICZ and D.V. PHILLIPS, "An Automatic Mesh Generation Scheme for Plane and Curved Surface by Isoparametric Coordinates", International Journal for Numerical Methods in Engineering, Vol 3 (1971).
20. G. STEINMUELLER, "Restrictions in the Application of Automatic Mesh Generation Schemes by 'Isoparametric' Coordinates", International Journal for Numerical Methods in Engineering, Vol 8 (1974).
21. J.C. CAVENDISH "Automatic Triangulation of Arbitrary Planar Domains for the Finite Element Method", International Journal for numerical methods in Engineering, Vol.8 (1974).
22. K.H. HUEBNER, "The Finite Element Method for Engineers", John Wiley & Sons, 1975.

23. C.S. DESAI, J.F. ABEL, "Introduction to the Finite Element Method", Van Nostrand Reinhold Company, 1972.
24. C.A. BREBBIA, A.J. FERRANTE, "Computational Methods for the Solution of Engineering Problems", Pentech Press Ltd. 1978.
25. C.S. DESAI, "Elementary Finite Element Method", Prentice Hall, Inc., 1979.
26. K.J. BATHE and E.L. WILSON, "Numerical Methods in Finite Element Analysis", Prentice Hall, Inc., 1976.
27. A. JENNINGS, "Matrix Computation for Engineers and Scientists", John Wiley & Sons, 1977.
28. M. LAFUENTE, M. CERROLAZA, "Ecuación de Campo Generalizada y Elasticidad Bidimensional por el Método de los Elementos Finitos", Tese de Graduação, Universidad Central de Venezuela, 1979.
29. F. KREITH, W.Z. Black, "Basic Heat Transfer", Haper & Row, Publishers, New York, 1980.
30. "The Finite Element Technique", edited by C.A. Brebbia, A. J. Ferrante, Universidade Federal do Rio Grande do Sul, Porto Alegre, 1975.
31. K.J. BATHE and M.R. KHOSHGOFTAAR, "Analysis of Nonlinear Heat Transfer and Field Problems", Mechanical Engineering Department, M.I.T, Massachusetts 02139.
32. J.D. KNUDSEN and D.L. KATZ, "Fluid Dynamics and Heat Transfer", Mc Graw-Hill Book Company, N.Y, 1978.
33. TIMOSHENKO G. "Theory of Elasticity, Mc Graw-Hill Book Company, 1972.

34. C.A. BREBBIA, S. WALKER and M. KAVANAGH, "Wave Oscillation Problems in Deep and Shallow Waters", Offshore Structures Engineering, Carneiro, F.L.L.B., A.J. Ferrante, C.A. Brebbia (editores), Pentech Press, 1979.
35. A.J. FERRANTE, "Fundamentos del Método de Elementos Finitos" Universidad Central de Venezuela, IMME, Caracas, 1979.
36. M. LAFUENTE, A.J. FERRANTE, A.GUELL, E. CASTILLA, "Um Sistema de Elementos Finitos para Minicomputador", II Colóquio Brasil-Venezuela de Engenharia - Anais - 1981.
37. ODEN, J.T., "Finite Element Applications in Nonlinear Structural Analysis", Proceedings of the Conference on Finite Element Methods, Vanderbilt University, Tennessee, 1969.
38. C. GENATIOS, A.J. FERRANTE, C.A. BREBBIA, "Programa Educacional de Minicomputador para Análisis de Redes Hidráulicas", Memoria Primer Colóquio Brasil-Venezuela de Ingeniería Civil, Junho, 2 a 6, 1980.

APÊNDICE

GLOSSÁRIO

(de termos de computador)

- ACESSO (memória): caracteriza a forma em que uma memória pode ser lida ou escrita.
- ACESSO DIRETO: tipo de memória na qual o acesso pode ser feito diretamente, a dados em qualquer direção.
- ACESSO SEQUENCIAL: tipo de memória na qual o acesso só pode ser feito na sequência em que foram armazenados os dados.
- ACUMULADOR: registro da unidade central no qual se executam as operações nas máquinas de uma direção.
- ÁLGEBRA DE BOOLE: Estrutura de tipo algébrico sobre variáveis que podem tomar dois valores distintos (Funções AND, OR, NOT).
- ALGOL: (ALGOrithmic Language) Linguagem de descrição de algoritmos empregada na comunicação de algoritmos e na programação de problemas científicos.
- ANTECIPAÇÃO: Técnica consistente de buscar na memória as instruções (e, eventualmente, suas operações) com antecipação, com o objetivo de evitar a unidade aritmética de um operador.
- ARITMÉTICA FIXA: Aritmética sobre números representados em ponto fixo (ver esta palavra).
- ARITMÉTICA FLUTUANTE: Aritmética sobre números representados em ponto flutuante.

- CAPACIDADE DE MEMÓRIA: Expressa-se em número de posições de memória, ou também em múltiplos de 210 posições de memória, unidade chamada de "K". Uma memória de 4K contém 4096 posições.
- CARACTER: elemento de informação de vários dígitos binários (em geral de 6 a 8) , considerado por alguns órgãos de ordenadores como unidade de informação.
- CARREGADOR (In loader): programa que permite armazenar em memória, um programa previamente montado ou compilado (tendo em conta possíveis mudanças de direção) ou vários módulos de programas montados ou compilados independentemente, levando em consideração os seus laços.
- CICLO DE MEMÓRIA: tempo mínimo entre duas operações sucessivas sobre um bloco de memória de acesso direto.
- CIRCUITO INTEGRADO: circuito fabricado em uma pastilha de silício que consta de alguns milhares de componentes fabricados mediante operações de mascaramento, de difusão de impurezas e de metalização.
- COMPILADOR: programa de computador que aceita linguagens copiáveis e os converte em instruções de máquina.
- COMPATIBILIDADE: capacidade que apresentam ordenadores diferentes para processar os mesmos programas.

gramas (compatibilidade de software) ou para representar as mesmas interfases com relação com os órgãos periféricos (compatibilidade de hardware).

- COMPUTADOR: máquina eletrônica para o processamento de grandes massas de informação codificada e capaz de realizar operações lógicas e matemáticas a alta velocidade.
- COMPUTADOR ANALÓGICO: um computador no qual as representações e operações são efetuadas de forma analógica, isto é, usando quantidades físicas variáveis.
- COMPUTADOR DIGITAL: computador capaz de realizar operações e representar dados em forma digital.
- COMPUTADOR MICROPROGRAMADO: computador cujas instruções se executam sob o controle de um microprograma.
- CONVERSACIONAL: adjetivo que qualifica a uma utilização do ordenador, em que o homem dialoga com a máquina com a ajuda de um terminal, tal como uma máquina de escrever com funções de entrada e saída, etc. Sinônimo: interativo.
- CPU: ver Processador.
- CRT, terminal: terminal construído baseado em tubos de raios catódicos (cathode Ray Tubes).
- DADOS: elementos ou "itens" conhecidos.
- DIREÇÃO: número que identifica um lugar da memória.
- DIGITAL: adjetivo que caracteriza as informações representadas sob forma discreta.

- DIREÇÃO ABSOLUTA: identificação numérica de cada célula da memória.
- DIRECIONAMENTO: operação de fornecimento de uma informação chamada direção, a qual permite selecionar um elemento entre um conjunto de elementos do mesmo tipo.
- DIRECIONAMENTO DIRETO: uma parte da direção da instrução que contém a direção do operando.
- DIRECIONAMENTO INDIRETO: uma parte da direção da instrução contém a direção da outra célula de memória.. Nesta está a direção do operando.
- DIRECIONAMENTO IMEDIATO: uma parte da direção da instrução contém o operando.
- EPROM: PROM apagável. (Erasable Programmable Read only Memory)
- ERRO: termo geral que se refere a qualquer desvio de uma quantidade medida com referência ao valor correto.
- FIRMWARE (Palavra inglesa): por analogia com o hardware (material) e software (programas), o firmware designa a microprogramação, que, num sentido estrito, não pertence completamente nem ao hardware nem ao software.
- FLIP-FLOP (Palavra inglesa): elemento de memória de uma posição binária, sistema de dois estados estáveis.
- FORTRAN (FÓrmula TRANslation): linguagem de programação cientificamente orientada, desenvolvida em 1957.
- HARDWARE: unidades físicas que constituem o computador.

- HEXADECIMAL: sistema de enumeração de base 16. Utiliza-se frequentemente para uma representação condensada do número binário.
- INFORMAÇÃO: conhecimento com significado para o ser humano. Pode ser extraída de um conjunto de dados.
- INTERATIVO: ver conversacional.
- INTERFASE: lugar onde dois sub-sistemas de um mesmo sistema informático interatuam; por extensão, dispositivo que permite associar dois conjuntos que não apresentam as mesmas especificações de interfase.
- INSTRUÇÃO: informação digital que indica a um computador uma operação particular a ser executada, assim como os dados implicados ou suas posições.
- INSTRUÇÃO DE MÁQUINA: conjunto de bits que dirigem ao computador na realização de alguma operação: soma, subtração, multiplicação, divisão, armazenamento, carregamento, leitura, escrita, etc.
- LINGUAGEM ASSEMBLER: linguagem simbólica de baixo nível, conversão direta da linguagem de máquina.
- LINGUAGEM COMPILADOR: linguagem simbólica de alto nível em que cada símbolo pode desencadear várias instruções de máquina.
- LINGUAGEM DE COMPUTADOR: conjunto de símbolos por meio do qual o usuário se comunica com o computador.
- LINGUAGEM DE MÁQUINA: conjunto de instruções de máquina as

quais são obedecidas diretamente pelo computador, sem necessidade de tradução.

- LINGUAGEM SIMBÓLICA: linguagem de programação, também chamada linguagem fonte, na qual os códigos de operação e os operandos podem ser designados por símbolos.
- LINHA ONIBUS: conjunto de fios que servem a um certo número de dispositivos digitais. A comunicação dos equipamentos através destes fios, impede qualquer outra, enquanto dura.
- MANTISA: na representação em ponto flutuante, número formado pelas cifras mais significativas do número por representar.
- MÁQUINA DE 1,2,3 DIREÇÕES: ordenadores classificados segundo o número de direções de posições de memória que figurem na instrução.
- MEIO: substância física sobre a qual se gravam os dados. Por exemplo: fita magnética, disco magnético, cartões perfurados, papel, etc.
- MEMÓRIA : sistema no qual se pode armazenar informações e recuperá-las mais tarde.
- MEMÓRIA AUXILIAR: suplemento da memória principal, usualmente baseada em discos magnéticos, tambores, fitas, etc.
- MEMÓRIA BUFFER: (em inglês: cache memory): memória de pequena capacidade e alta velocidade interposta como tampão entre a memória principal e a unidade central: carrega-se com arranjo de técnicas de antecipação implícitas (ver antecipação).

- MEMORIA CORE: aneis de metal que são capazes de assumir dois estados diferentes, podendo representar as quantidades 0 ou 1.
- MEMORIA DE BORBULHA: chip em forma de borbulha com capacidade para 10-megabits.
- MEMÓRIA DE MASSA: memória auxiliar de acesso por blocos ou aleatório (discos, tambores, lâminas magnéticas).
- MEMÓRIA LOCAL: memória de pequena capacidade e rápido acesso que se utiliza na unidade central.
- MEMÓRIA PRINCIPAL: memória interna do computador, a partir da qual se executam as instruções.
- MEMÓRIA VIRTUAL: é o espaço em memória definido por uma faixa de direções especificada pelo programador e diferente das direções utilizadas pelo sistema. Permite considerar a memória interna como ilimitada ou somente limitada pela capacidade de direcionamento do computador.
- MICROCIRCUITOS: circuitos minúsculos nos quais todos os componentes são formados quimicamente sobre uma única peça de material. Contém o equivalente de centenas e inclusive milhares de transistores e outros componentes.
- MICROCOMPUTADOR: pequeno computador de baixo custo. Contém pelo menos um microprocessador, memória (ROM, PROM, EPROM ou RAM) e um circuito de input e output.
- MICROLINGUAGEM: linguagem de microprogramação.
- MICROPROCESSADOR: unidade de processamento central contida em

um "chip". É o coração de um microcomputador.

- MICROPROGRAMAÇÃO: designa uma técnica de realização do sequenciador central de um ordenador, na qual cada instrução é interpretada e executada sob o controle de um programa chamado microprograma, formado por uma sucessão de micro-instruções, cada uma das quais governa uma fase da execução da instrução (ver Firmware e modelo de Wilkes).
- MICROSEGUNDO: um milionésimo de segundo.
- MILESEGUNDO: um milhésimo de segundo.
- MINICOMPUTADOR: computador de baixo custo introduzido no mercado por volta de 1965.
- MODELO DE WILKES: primeiro modelo de máquina microprogramada. Serve de referência para qualquer estudo da microprogramação.
- MONOPROCESSADOR: Ordenador que não possui mais que uma unidade central.
- MULTIPROCESSADOR: ordenador que possui várias unidades centrais.
- NANOSEGUNDO: 10^{-9} segundos.
- OCTAL: sistema de numeração de base 8. Utilizado frequentemente como representação condensada de números binários.
- OPERAÇÃO: ação definida por uma parte de uma instrução ao computador.
- OPERADOR: é todo o dispositivo que efetua uma opera -

ção, um processamento: operador aritimético, operador lógico, etc. Também todo sinal que define uma operação por efetuar.

- OPERANDO: quantidade em uma instrução (resultado, argumento ou número), a parte direcionante de uma instrução.

- ORDENADOR: nome dado aos calculadores eletrônicos digitais.

- PÁGINA: unidade de memória de tamanho fixo, usualmente de 2048 ou 4096 bytes.

- PAGINAÇÃO: método de utilização de uma memória principal dividida em páginas. A paginação pode permitir: a) direcionar uma memória cuja capacidade não é congruente com a capacidade de direcionamento da instrução; 2) não carregar os programas de um só bloco; 3) gestionar dinamicamente a memória não carregando em um instante dado mais do que aquelas páginas efetivamente úteis no programa.

- PALAVRA: conjunto de caracteres que ocupa um lugar na memória e pode ser tratado como uma unidade. Utiliza-se somente se o elemento de informação é superior a 8 dígitos.

- PARIDADE: controle da validade de uma informação depois da sua transmissão, armazenando o processamento, mediante a adição de um bit (bit de paridade), posicionado antes da operação e testado depois dela. O bit de paridade faz par (paridade par) ou impar (paridade impar) a soma dos bits de informação. A aparição de uma paridade anormal corresponde a um erro.

- PARTIÇÃO: termo geralmente empregado para designar uma parte de um todo que por si só constitui um conjunto lógico.
- PLOTTER DIGITAL: equipamento que esboça gráficos sob a direção de um computador.
- PROCESSADOR: (unidade central de processamento) o nervo central de qualquer sistema de computador digital, o qual controla e coordena as atividades das outras partes do computador, realizando também os processos aritméticos e lógicos aplicados aos dados. Compõe-se de três partes lógicas: unidade aritmética, unidade de memória e unidade de controle.
- PROCESSADOR CENTRAL: equivalente à unidade central.
- PROCESSADOR DE ENTRADA-SAÍDA : equivalente a canal.
- PROCESSAMENTO DE DADOS: - operações efetuadas nos dados, usualmente por equipamento automático, para extrair alguma informação.
- PROCESSO: termo genérico que compreende várias operações, como computar, imprimir dados, manejar dados, etc.
- PROGRAMA: sequência significativa de instruções.
- PROGRAMA FONTE: programa escrito em linguagem simbólica.
- PROGRAMA OBJETO: programa em linguagem de máquina, diretamente executado pelo computador.
- PROM (programável ROM): memória com base em chips, programada para conservar a informação mesmo que seja a pagada do computador.

- PROTEÇÃO DE MEMÓRIA: técnica para evitar que erros cometidos por um usuário destruam o conteúdo de zonas de memórias que não foram para ele reservadas.
- PUNTERO: registro de célula de memória de direção de terminada que contém a direção de uma informação ou de um conjunto de informações.
- PONTO FIXO: ver vírgula Fixa.
- PONTO FLUTUANTE: ver vírgula Flutuante.
- RAM (Random access memory): memória para escrever e/ou ler, mas cujo conteúdo é volátil, perde-se quando se apaga o computador.
- RECORD: grupo de dados relacionados, tratados como uma unidade.
- REGISTRO: ver Record.
- REGISTROS FÍSICOS: unidades de transmissão ou de armazenamento sobre memórias auxiliares.
- REGISTROS LÓGICOS: conhecidos do programador que trabalha em linguagem evoluída, e que estão se for necessário, divididos em registros físicos.
- ROM (read only memory) : não é possível escrever dados em memória tipo ROM.
- SEQUENCIADOR: órgão de governo de um operador sequencial, ou de um ordenador. Distribui em instantes precisos, as ordens que posicionam os circuitos do operador ou do ordenador visando executar as diferentes fases da operação ou da instrução.

- SEGMENTAÇÃO: método de programação que permite fracionar os programas demasiado longos, que não caberiam totalmente na memória principal em elementos chamados segmentos, que irão sendo chamados à memória de acordo com as necessidades.
- SISTEMA BINÁRIO: sistema numérico que utiliza a base dos (Dígitos 0 ou 1).
- SISTEMA OCTAL: sistema numérico com base 8 (dígitos de 0 a 7).
- SOFTWARE: programas que podem ser usados no computador. Distinguem-se o software de base e o de aplicação.
- TECNOLOGIA: disciplina da informática a base de eletrônica e mecânica.
- TERMINAL: órgão periférico, geralmente situado a distância do ordenador, que permite a comunicação entre o homem e ele. Exemplo: tele-impressor, terminal CRT, sistema com entrada de cartões e saída por impressoras, etc.
- TEMPO COMPARTIDO (Sistema de): sistema mediante o qual mais de um usuário podem compartilhar do computador (tempo de processador) através de vários terminais remotos.
- TEMPO DE ACESSO: tempo que utiliza o computador para procurar e transferir instruções providas da memória e em direção a ela.
- TEMPO REAL: tempo que se refere a processos físicos reais.

- TRACK: canal em um equipamento de memória magnética para armazenamento de dados.
- TRADUTOR: programa de computador que converte um programa de uma linguagem de computador a outro, por exemplo. de uma linguagem fonte a objeto.
- TRANSISTOR: equipamento eletrônico em estado sólido capaz de realizar as mesmas funções que válvulas, sendo menores, baratos, requerendo menos energia e gerando menos calor.
- UNIBUS: bus único, ao qual se conectam todos os elementos componentes de um ordenador : unidade central, memória, canais, etc. utilizado em alguns computadores pequenos.
- UNIDADE ARITMÉTICA LÓGICA: executa as operações aritméticas e lógicas.
- UNIDADE CENTRAL: ver processador.
- VÁLVULA: circuito eletrônico utilizado nos primeiros computadores.
- VARIÁVEL: em informática, trata-se de uma informação identificada por um nome e uma direção que pode assumir um valor (ou conjunto de valores) em um domínio dado, no curso de desenvolvimento de um programa.
- VARIÁVEL LÓGICA: variável da álgebra de Boole que pode assumir valores de 0 a 1.

- VÍRGULA FIXA: representação dos números expressados em um sistema de numeração, com a ajuda de um número fixo de dígitos, no qual o ponto que separa a parte inteira da parte fracionada, estão em um lugar fixo em relação a um dos extremos do número representado.
- VIRGULA FLUTUANTE: a posição da vírgula não está fixa com relação a um dos extremos do número representado.
- VOLATILIDADE: qualifica uma memória, cujo conteúdo se perde em caso de corte da alimentação elétrica.